VISUAL AGI-KBI HYPERKB OS V3

Continuation and expansion of what V1/V2 did not fully cover: image generation, video generation, commercial ads, non-ordinary conceptual art, pop/flat/minimal/patchwork design, Remotion/HyperFrames, UGC realism, audio-visual production boxes, HyperPrompt loops, evaluation courts and GraphRAG creative memory.

Author: Zouhair Moussaif /

Epistemic Safety Declaration

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-002; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

What V3 Adds Beyond V1/V2

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-003; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Ontological Constitution of Visual KBI

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-004; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Central Research Question

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-005; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Generative Kernel

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-006; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Non-Ordinary Symbol Transfer

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-007; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Moussaif Aesthetic Compression

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-008; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Color-Text Governance

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-009; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Typography as Cognitive Routing

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-010; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Minimal Design Without Emptiness

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-011; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Flat Design as Operational Abstraction

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-012; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Pop Art as Attention Weapon

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-013; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Patchwork as Nonlinear Knowledge Map

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-014; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Conceptual Art for Ads

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-015; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Commercial Concept Truth

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=editorial grayscale with electric
accent. Keep the main text huge and legible. Generate 4 variants, then
reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-016; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Visual Weirdness With Conversion Discipline

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-017; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Prompt-to-Layout DSL

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-018; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Prompt-to-Shot DSL

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-019; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Prompt-to-Motion DSL

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-020; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

High-End Production Boxes

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-021; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Agency Pipeline Constitution

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-022; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Creative Director Agent

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-023; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Semiotic Engineer Agent

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-024; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Model Router Agent

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-025; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Video Director Agent

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-026; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

UGC Authenticity Agent

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-027; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Audio Identity Agent

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-028; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Motion Graphics Agent

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-029; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Evaluation Judge Agent

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-030; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

RAG Chunking for Visual Briefs

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=minimal cream/black/red. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-031; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

GraphRAG for Brand Memory

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-032; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Asset Versioning System

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-033; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Style Token Registry

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-034; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Shot Token Registry

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-035; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Color Token Registry

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-036; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Typography Token Registry

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-037; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Rights and Provenance Registry

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-038; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Consent and Likeness Registry

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-039; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Model Availability Watch

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=editorial grayscale with electric
accent. Keep the main text huge and legible. Generate 4 variants, then
reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-040; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Benchmark-Inspired QA

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-041; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

VBench-Inspired Video Metrics

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-042; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

T2I-CompBench-Inspired Image Metrics

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-043; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

T2V-CompBench-Inspired Motion Metrics

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-044; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

MLLM Judge Calibration

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-045; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Human Creative Review Protocol

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-046; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Commercial Ad Review Court

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-047; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Hallucination and Fake Proof Court

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-048; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

False Novelty Court

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-049; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Reductionism Court

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-050; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Weak Operationalization Court

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-051; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Agentic Failure Court

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-052; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Prompt Injection Safety

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-053; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

HyperPrompt Loop Stack

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-054; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Creative Memory Update Protocol

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=minimal cream/black/red. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-055; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Versioning and Deprecation

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNE

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTIO

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=pop-art cyan/yellow/magenta. Keep
the main text huge and legible. Generate 4 variants, then reject
generic ones.
```

DESTRUCTION COUR

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-056; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Deployment Modes

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

:
. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=patchwork earth/neon. Keep the main
text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-057; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Research Program

CLAIM / FUNCTION

V3 treats every asset as an operational knowledge unit: it has a brief, audience state, style logic, production method, evaluation court, rights state, and memory update.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: client ad goal -> audience friction -> proof claim -> image/video grammar -> platform constraint -> QA court -> launch or reject.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an unusual visual object. Use palette=editorial grayscale with electric accent. Keep the main text huge and legible. Generate 4 variants, then reject generic ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-058; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

AGI-Readiness Metrics

CLAIM / FUNCTION

The Moussaif principle here is simple: no visual style without function, no function without mechanism, no mechanism without a test, no test without repair memory.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: brand memory -> forbidden clichés -> allowed styles -> prompt DSL -> model router -> post-production -> provenance log.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=luxury black/gold/white. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved. RAG metadata: chunk_id=V3-FND-059; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Final Pre-Model Kernel

CLAIM / FUNCTION

This page closes a missing layer: the OS must not only generate images and videos; it must govern meaning transfer, model routing, evidence rank, commercial truth, and iterative repair.

OPERATIONAL KERNEL

. Operationally, the page converts a missing dimension into a reusable production unit. It belongs in the KB only if it can change a prompt, a shot list, a layout, an evaluation checklist, or a deployment decision.

MECHANISM

Input: raw idea -> extract active difference -> choose visual metaphor -> choose model/tool -> generate variants -> judge -> repair -> store tokens.

HYPERPROMPT INJECTION

```
Brief: [idea]. Extract the active difference. Translate it into an
unusual visual object. Use palette=neo-tech violet/cyan/gold. Keep the
main text huge and legible. Generate 4 variants, then reject generic
ones.
```

DESTRUCTION COURT

Reject if it is merely decorative, if text is unreadable, if the metaphor has no commercial function, if it hides uncertainty, or if rights/consent are unresolved.
RAG metadata: chunk_id=V3-FND-060; confidence=theoretical-synthesis; update_status=active; model_dependency=none; risk=medium

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

model-atlas image positioning-&-ontology

GPT Image 2 / OpenAI Image API - Positioning & Ontology

CLAIM / FUNCTION

GPT Image 2 / OpenAI Image API is treated as a bounded production organ, not a universal answer. Type: image generation and editing. Best use: semantic layouts, ad concepts, image edits, text-aware commercial compositions. Epistemic rank: Supported, current-docs dependent.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to GPT Image 2 / OpenAI Image API only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-061; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

GPT Image 2 / OpenAI Image API - Commercial Use Map

CLAIM / FUNCTION

Commercially, GPT Image 2 / OpenAI Image API must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For GPT Image 2 / OpenAI Image API, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for GPT Image 2 / OpenAI Image API: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-062; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image aesthetic-grammar

GPT Image 2 / OpenAI Image API - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for GPT Image 2 / OpenAI Image API maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-063; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image prompt-injection-loop

GPT Image 2 / OpenAI Image API - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for GPT Image 2 / OpenAI Image API is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For GPT Image 2 / OpenAI Image API: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-064; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image shot/motion-grammar

GPT Image 2 / OpenAI Image API - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for GPT Image 2 / OpenAI Image API converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for GPT Image 2 / OpenAI Image API: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-065; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image pipeline-&-toolchain

GPT Image 2 / OpenAI Image API - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for GPT Image 2 / OpenAI Image API defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: availability, text fidelity, brand/legal review, synthetic-media disclosure.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using GPT Image 2 / OpenAI Image API: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-066; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

GPT Image 2 / OpenAI Image API - Evaluation Court

CLAIM / FUNCTION

Evaluation court for GPT Image 2 / OpenAI Image API uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of GPT Image 2 / OpenAI Image API on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-067; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image failure-modes-&-repair

GPT Image 2 / OpenAI Image API - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for GPT Image 2 / OpenAI Image API must be known before deployment: availability, text fidelity, brand/legal review, synthetic-media disclosure.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on GPT Image 2 / OpenAI Image API output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-068; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

GPT Image 2 / OpenAI Image API - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for GPT Image 2 / OpenAI Image API stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a GPT Image 2 / OpenAI Image API
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-069; model=GPT Image 2 / OpenAI Image API; confidence=Supported, current-docs dependent; update_status=watch; risk=availability, text fidelity, brand/legal review, synthetic-media disclosure

model-atlas image positioning-&-ontology

GPT Image 1 compatibility slot - Positioning & Ontology

CLAIM / FUNCTION

GPT Image 1 compatibility slot is treated as a bounded production organ, not a universal answer. Type: image generation fallback. Best use: fallback pipelines, existing integrations, lower-risk compatibility testing. Epistemic rank: Established as legacy/fallback.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to GPT Image 1 compatibility slot only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-070; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - Commercial Use Map

CLAIM / FUNCTION

Commercially, GPT Image 1 compatibility slot must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For GPT Image 1 compatibility slot, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for GPT Image 1 compatibility slot: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-071; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for GPT Image 1 compatibility slot maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-072; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for GPT Image 1 compatibility slot is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For GPT Image 1 compatibility slot: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-073; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for GPT Image 1 compatibility slot converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for GPT Image 1 compatibility slot: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-074; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

model-atlas image pipeline-&-toolchain

GPT Image 1 compatibility slot - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for GPT Image 1 compatibility slot defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: do not treat legacy capacity as frontier; compare against newer GPT Image models.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using GPT Image 1 compatibility slot: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-075; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - Evaluation Court

CLAIM / FUNCTION

Evaluation court for GPT Image 1 compatibility slot uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of GPT Image 1 compatibility slot on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-076; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

model-atlas image failure-modes-&-repair

GPT Image 1 compatibility slot - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for GPT Image 1 compatibility slot must be known before deployment: do not treat legacy capacity as frontier; compare against newer GPT Image models.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on GPT Image 1 compatibility slot output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-077; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

GPT Image 1 compatibility slot - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for GPT Image 1 compatibility slot stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a GPT Image 1 compatibility slot
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-078; model=GPT Image 1 compatibility slot; confidence=Established as legacy/fallback; update_status=watch; risk=do not treat legacy capacity as frontier; compare against newer GPT Image models

Sora 2 availability-watch - Positioning & Ontology

CLAIM / FUNCTION

Sora 2 availability-watch is treated as a bounded production organ, not a universal answer. Type: video and audio generation. Best use: physically-aware concept tests, cinematic motion, synchronized sound/dialogue prototypes. Epistemic rank: Supported with availability caveat.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Sora 2 availability-watch only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-079; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Commercial Use Map

CLAIM / FUNCTION

Commercially, Sora 2 availability-watch must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Sora 2 availability-watch, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Sora 2 availability-watch: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-080; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Sora 2 availability-watch maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-081; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Sora 2 availability-watch is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Sora 2 availability-watch: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-082; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Sora 2 availability-watch converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Sora 2 availability-watch: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-083; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Sora 2 availability-watch defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: product/API availability and regional access must be verified before client promises.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Sora 2 availability-watch: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-084; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Sora 2 availability-watch uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Sora 2 availability-watch on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-085; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Sora 2 availability-watch must be known before deployment: product/API availability and regional access must be verified before client promises.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Sora 2 availability-watch output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-086; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Sora 2 availability-watch - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Sora 2 availability-watch stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Sora 2 availability-watch generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-087; model=Sora 2 availability-watch; confidence=Supported with availability caveat; update_status=watch; risk=product/API availability and regional access must be verified before client promises

Google Imagen 4 - Positioning & Ontology

CLAIM / FUNCTION

Google Imagen 4 is treated as a bounded production organ, not a universal answer. Type: image generation. Best use: hero images, high clarity, campaign ideation, typography-sensitive image variants. Epistemic rank: Supported.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Google Imagen 4 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-088; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Google Imagen 4 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Google Imagen 4, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Google Imagen 4: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-089; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Google Imagen 4 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-090; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Google Imagen 4 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Google Imagen 4: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-091; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Google Imagen 4 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Google Imagen 4: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-092; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Google Imagen 4 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: text and brand marks still need human QA; plan limits vary.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Google Imagen 4: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-093; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Google Imagen 4 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Google Imagen 4 on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-094; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Google Imagen 4 must be known before deployment: text and brand marks still need human QA; plan limits vary.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Google Imagen 4 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-095; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

Google Imagen 4 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Google Imagen 4 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Google Imagen 4 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-096; model=Google Imagen 4; confidence=Supported; update_status=watch; risk=text and brand marks still need human QA; plan limits vary

model-atlas video positioning-&-ontology

Google Veo 3.1 - Positioning & Ontology

CLAIM / FUNCTION

Google Veo 3.1 is treated as a bounded production organ, not a universal answer. Type: video generation with native audio. Best use: short ad clips, image-directed motion, social video, cinematic b-roll with sound. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Google Veo 3.1 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-097; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Google Veo 3.1 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Google Veo 3.1, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Google Veo 3.1: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-098; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Google Veo 3.1 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-099; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Google Veo 3.1 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNE

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
[HYPERPROMPT] For Google Veo 3.1: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-100; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Google Veo 3.1 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Google Veo 3.1: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-101; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Google Veo 3.1 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: duration, resolution and API features vary by surface and plan.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Google Veo 3.1: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-102; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Google Veo 3.1 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Google Veo 3.1 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-103; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

Google Veo 3.1 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Google Veo 3.1 must be known before deployment: duration, resolution and API features vary by surface and plan.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Google Veo 3.1 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-104; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

model-atlas video rag/graphrag-metadata

Google Veo 3.1 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Google Veo 3.1 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Google Veo 3.1 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-105; model=Google Veo 3.1; confidence=Supported; update_status=watch; risk=duration, resolution and API features vary by surface and plan

model-atlas cost/scale positioning-&-ontology

Google Veo 3.1 Lite/Fast - Positioning & Ontology

CLAIM / FUNCTION

Google Veo 3.1 Lite/Fast is treated as a bounded production organ, not a universal answer. Type: cost/scale video generation. Best use: high-volume video applications, rapid experiments, cheaper variant pipelines. Epistemic rank: Supported, cost-version dependent.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Google Veo 3.1 Lite/Fast only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-106; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Commercial Use Map

CLAIM / FUNCTION

Commercially, Google Veo 3.1 Lite/Fast must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Google Veo 3.1 Lite/Fast, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Google Veo 3.1 Lite/Fast: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-107; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Google Veo 3.1 Lite/Fast maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-108; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Google Veo 3.1 Lite/Fast is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNE

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
[HYPERPROMPT] For Google Veo 3.1 Lite/Fast: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-109; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Google Veo 3.1 Lite/Fast converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Google Veo 3.1 Lite/Fast: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-110; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Google Veo 3.1 Lite/Fast defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: quality trade-offs must be measured on your own shots.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Google Veo 3.1 Lite/Fast: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-111; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Google Veo 3.1 Lite/Fast uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Google Veo 3.1 Lite/Fast on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-112; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Google Veo 3.1 Lite/Fast must be known before deployment: quality trade-offs must be measured on your own shots.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Google Veo 3.1 Lite/Fast output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-113; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Google Veo 3.1 Lite/Fast - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Google Veo 3.1 Lite/Fast stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Google Veo 3.1 Lite/Fast generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-114; model=Google Veo 3.1 Lite/Fast; confidence=Supported, cost-version dependent; update_status=watch; risk=quality trade-offs must be measured on your own shots

Runway Gen-4.5 - Positioning & Ontology

CLAIM / FUNCTION

Runway Gen-4.5 is treated as a bounded production organ, not a universal answer. Type: frontier video generation. Best use: cinematic photorealism, high-fidelity product shots, visual-control tests. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Runway Gen-4.5 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-115; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Runway Gen-4.5 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Runway Gen-4.5, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Runway Gen-4.5: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-116; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

model-atlas frontier aesthetic-grammar

Runway Gen-4.5 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Runway Gen-4.5 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-117; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Runway Gen-4.5 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNE

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
[HYPERPROMPT] For Runway Gen-4.5: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-118; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Runway Gen-4.5 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Runway Gen-4.5: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-119; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Runway Gen-4.5 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: object permanence and causal sequencing still require shot-level QA.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Runway Gen-4.5: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-120; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Runway Gen-4.5 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Runway Gen-4.5 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-121; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Runway Gen-4.5 must be known before deployment: object permanence and causal sequencing still require shot-level QA.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Runway Gen-4.5 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-122; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4.5 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Runway Gen-4.5 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Runway Gen-4.5 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-123; model=Runway Gen-4.5; confidence=Supported; update_status=watch; risk=object permanence and causal sequencing still require shot-level QA

Runway Gen-4 - Positioning & Ontology

CLAIM / FUNCTION

Runway Gen-4 is treated as a bounded production organ, not a universal answer. Type: reference-controlled video. Best use: consistent subject/object shots, 5-10 second clips, advertising prototypes. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Runway Gen-4 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-124; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

model-atlas reference-controlled commercial-use-map

Runway Gen-4 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Runway Gen-4 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Runway Gen-4, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Runway Gen-4: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-125; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

model-atlas reference-controlled aesthetic-grammar

Runway Gen-4 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Runway Gen-4 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-126; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

Runway Gen-4 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Runway Gen-4 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Runway Gen-4: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-127; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

model-atlas reference-controlled shot/motion-grammar

Runway Gen-4 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Runway Gen-4 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Runway Gen-4: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-128; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

Runway Gen-4 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Runway Gen-4 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: break complex narratives into shots to reduce drift.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Runway Gen-4: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-129; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

model-atlas reference-controlled evaluation-court

Runway Gen-4 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Runway Gen-4 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Runway Gen-4 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-130; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

Runway Gen-4 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Runway Gen-4 must be known before deployment: break complex narratives into shots to reduce drift.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Runway Gen-4 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-131; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

Runway Gen-4 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Runway Gen-4 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Runway Gen-4 generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-132; model=Runway Gen-4; confidence=Supported; update_status=watch; risk=break complex narratives into shots to reduce drift

model-atlas image positioning-&-ontology

Midjourney V7 - Positioning & Ontology

CLAIM / FUNCTION

Midjourney V7 is treated as a bounded production organ, not a universal answer. Type: image generation and art direction. Best use: moodboards, pop art, editorial surrealism, patchwork, concept art ads. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Midjourney V7 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-133; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Midjourney V7 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Midjourney V7, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Midjourney V7: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-134; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

model-atlas image aesthetic-grammar

Midjourney V7 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Midjourney V7 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-135; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Midjourney V7 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Midjourney V7: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-136; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Midjourney V7 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Midjourney V7: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-137; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Midjourney V7 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: final typography and factual claims need post-design QA.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Midjourney V7: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-138; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Midjourney V7 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Midjourney V7 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-139; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Midjourney V7 must be known before deployment: final typography and factual claims need post-design QA.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Midjourney V7 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-140; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

Midjourney V7 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Midjourney V7 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Midjourney V7 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-141; model=Midjourney V7; confidence=Supported; update_status=watch; risk=final typography and factual claims need post-design QA

model-atlas reference-control positioning-&-ontology

Midjourney Omni Reference - Positioning & Ontology

CLAIM / FUNCTION

Midjourney Omni Reference is treated as a bounded production organ, not a universal answer. Type: reference-control system. Best use: stabilizing characters, objects, style memories and branded visual motifs. Epistemic rank: Supported by official docs.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Midjourney Omni Reference only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-142; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Commercial Use Map

CLAIM / FUNCTION

Commercially, Midjourney Omni Reference must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Midjourney Omni Reference, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Midjourney Omni Reference: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-143; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Midjourney Omni Reference maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-144; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Midjourney Omni Reference is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Midjourney Omni Reference: Extract Δ. Translate Δ
into one visual paradox. Set typography hierarchy. Generate. Judge with
5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-145; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Midjourney Omni Reference converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Midjourney Omni Reference: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-146; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

model-atlas reference-control pipeline-&-toolchain

Midjourney Omni Reference - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Midjourney Omni Reference defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: reference does not solve rights; owned/licensed references only.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Midjourney Omni Reference: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-147; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Midjourney Omni Reference uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Midjourney Omni Reference on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-148; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Midjourney Omni Reference must be known before deployment: reference does not solve rights; owned/licensed references only.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Midjourney Omni Reference output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-149; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Midjourney Omni Reference - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Midjourney Omni Reference stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Midjourney Omni Reference generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-150; model=Midjourney Omni Reference; confidence=Supported by official docs; update_status=watch; risk=reference does not solve rights; owned/licensed references only

Ideogram 4.0 - Positioning & Ontology

CLAIM / FUNCTION

Ideogram 4.0 is treated as a bounded production organ, not a universal answer. Type: open-weight typography-first image model. Best use: posters, packaging, slogans, multi-line text, multilingual design scenes. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Ideogram 4.0 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-151; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Ideogram 4.0 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Ideogram 4.0, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Ideogram 4.0: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-152; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Ideogram 4.0 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-153; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Ideogram 4.0 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Ideogram 4.0: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-154; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Ideogram 4.0 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Ideogram 4.0: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-155; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Ideogram 4.0 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: commercial license and local deployment requirements must be checked.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Ideogram 4.0: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-156; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Ideogram 4.0 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Ideogram 4.0 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-157; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

model-atlas open-weight failure-modes-&-repair

Ideogram 4.0 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Ideogram 4.0 must be known before deployment: commercial license and local deployment requirements must be checked.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Ideogram 4.0 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-158; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

Ideogram 4.0 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Ideogram 4.0 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Ideogram 4.0 generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-159; model=Ideogram 4.0; confidence=Supported; update_status=watch; risk=commercial license and local deployment requirements must be checked

model-atlas structured positioning-&-ontology

Ideogram JSON/BBox prompting - Positioning & Ontology

CLAIM / FUNCTION

Ideogram JSON/BBox prompting is treated as a bounded production organ, not a universal answer. Type: structured layout control. Best use: bounding-box text/object placement, disciplined graphic design, template outputs. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Ideogram JSON/BBox prompting only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-160; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Commercial Use Map

CLAIM / FUNCTION

Commercially, Ideogram JSON/BBox prompting must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Ideogram JSON/BBox prompting, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Ideogram JSON/BBox prompting: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-161; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Ideogram JSON/BBox prompting maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-162; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Ideogram JSON/BBox prompting is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Ideogram JSON/BBox prompting: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-163; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Ideogram JSON/BBox prompting converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Ideogram JSON/BBox prompting: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-164; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Ideogram JSON/BBox prompting defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: structured prompts still need visual QA and crop safety.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Ideogram JSON/BBox prompting: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-165; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Ideogram JSON/BBox prompting uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Ideogram JSON/BBox prompting on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-166; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Ideogram JSON/BBox prompting - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Ideogram JSON/BBox prompting must be known before deployment: structured prompts still need visual QA and crop safety.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Ideogram JSON/BBox prompting output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-167; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

model-atlas structured rag/graphrag-metadata

Ideogram JSON/BBox prompting - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Ideogram JSON/BBox prompting stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Ideogram JSON/BBox prompting generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-168; model=Ideogram JSON/BBox prompting; confidence=Supported; update_status=watch; risk=structured prompts still need visual QA and crop safety

Adobe Firefly Image / Studio - Positioning & Ontology

CLAIM / FUNCTION

Adobe Firefly Image / Studio is treated as a bounded production organ, not a universal answer. Type: creative suite image generation/editing. Best use: brand-safe enterprise work, moodboards, composites, Adobe-native production. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Adobe Firefly Image / Studio only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-169; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Commercial Use Map

CLAIM / FUNCTION

Commercially, Adobe Firefly Image / Studio must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Adobe Firefly Image / Studio, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Adobe Firefly Image / Studio: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-170; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Adobe Firefly Image / Studio maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-171; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Adobe Firefly Image / Studio is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Adobe Firefly Image / Studio: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-172; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Adobe Firefly Image / Studio converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Adobe Firefly Image / Studio: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-173; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Adobe Firefly Image / Studio defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: commercial-safety claims must be read in current terms.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Adobe Firefly Image / Studio: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-174; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Adobe Firefly Image / Studio uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Adobe Firefly Image / Studio on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-175; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Adobe Firefly Image / Studio must be known before deployment: commercial-safety claims must be read in current terms.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Adobe Firefly Image / Studio output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-176; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

Adobe Firefly Image / Studio - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Adobe Firefly Image / Studio stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Adobe Firefly Image / Studio generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-177; model=Adobe Firefly Image / Studio; confidence=Supported; update_status=watch; risk=commercial-safety claims must be read in current terms

model-atlas video positioning-&-ontology

Adobe Firefly Video / Editor / Sound - Positioning & Ontology

CLAIM / FUNCTION

Adobe Firefly Video / Editor / Sound is treated as a bounded production organ, not a universal answer. Type: video generation and editing suite. Best use: generative b-roll, video gaps, soundtrack/speech, timeline-first creative editing. Epistemic rank: Supported.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Adobe Firefly Video / Editor / Sound only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-178; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

Adobe Firefly Video / Editor / Sound - Commercial Use Map

CLAIM / FUNCTION

Commercially, Adobe Firefly Video / Editor / Sound must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Adobe Firefly Video / Editor / Sound, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Adobe Firefly Video / Editor / Sound: 1 big
title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1
color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-179; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

Adobe Firefly Video / Editor / Sound - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Adobe Firefly Video / Editor / Sound maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-180; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

model-atlas video prompt-injection-loop

Adobe Firefly Video / Editor / Sound - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Adobe Firefly Video / Editor / Sound is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Adobe Firefly Video / Editor / Sound: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-181; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

Adobe Firefly Video / Editor / Sound - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Adobe Firefly Video / Editor / Sound converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Adobe Firefly Video / Editor / Sound: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-182; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

model-atlas video pipeline-&-toolchain

Adobe Firefly Video / Editor / Sound - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Adobe Firefly Video / Editor / Sound defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: do not skip normal editor review; verify rights and audio license.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Adobe Firefly Video / Editor / Sound:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-183; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

model-atlas video evaluation-court

Adobe Firefly Video / Editor / Sound - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Adobe Firefly Video / Editor / Sound uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Adobe Firefly Video / Editor / Sound on 10
criteria. For each weak point, write a repair instruction and decide:
regenerate, edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-184; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

model-atlas video failure-modes-&-repair

Adobe Firefly Video / Editor / Sound - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Adobe Firefly Video / Editor / Sound must be known before deployment: do not skip normal editor review; verify rights and audio license.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Adobe Firefly Video / Editor / Sound output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-185; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

model-atlas video rag/graphrag-metadata

Adobe Firefly Video / Editor / Sound - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Adobe Firefly Video / Editor / Sound stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a Adobe Firefly Video / Editor / Sound
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-186; model=Adobe Firefly Video / Editor / Sound; confidence=Supported; update_status=watch; risk=do not skip normal editor review; verify rights and audio license

FLUX.2 dev/klein - Positioning & Ontology

CLAIM / FUNCTION

FLUX.2 dev/klein is treated as a bounded production organ, not a universal answer. Type: image generation/editing/combination. Best use: open/custom workflows, fast iteration, multi-image editing, local pipelines. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to FLUX.2 dev/klein only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-187; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

FLUX.2 dev/klein - Commercial Use Map

CLAIM / FUNCTION

Commercially, FLUX.2 dev/klein must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For FLUX.2 dev/klein, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for FLUX.2 dev/klein: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-188; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

FLUX.2 dev/klein - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for FLUX.2 dev/klein maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-189; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

FLUX.2 dev/klein - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for FLUX.2 dev/klein is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For FLUX.2 dev/klein: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-190; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

model-atlas image shot/motion-grammar

FLUX.2 dev/klein - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for FLUX.2 dev/klein converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for FLUX.2 dev/klein: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-191; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

model-atlas image pipeline-&-toolchain

FLUX.2 dev/klein - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for FLUX.2 dev/klein defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: compute, model license, and variant choice change commercial feasibility.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using FLUX.2 dev/klein: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-192; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

FLUX.2 dev/klein - Evaluation Court

CLAIM / FUNCTION

Evaluation court for FLUX.2 dev/klein uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of FLUX.2 dev/klein on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-193; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

model-atlas image failure-modes-&-repair

FLUX.2 dev/klein - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for FLUX.2 dev/klein must be known before deployment: compute, model license, and variant choice change commercial feasibility.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on FLUX.2 dev/klein output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-194; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

FLUX.2 dev/klein - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for FLUX.2 dev/klein stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a FLUX.2 dev/klein generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-195; model=FLUX.2 dev/klein; confidence=Supported; update_status=watch; risk=compute, model license, and variant choice change commercial feasibility

model-atlas in-context positioning-&-ontology

FLUX.1 Kontext - Positioning & Ontology

CLAIM / FUNCTION

FLUX.1 Kontext is treated as a bounded production organ, not a universal answer. Type: in-context image generation and editing. Best use: multi-turn edits, reference preservation, local/global edits, product variants. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to FLUX.1 Kontext only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-196; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

FLUX.1 Kontext - Commercial Use Map

CLAIM / FUNCTION

Commercially, FLUX.1 Kontext must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For FLUX.1 Kontext, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for FLUX.1 Kontext: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-197; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

model-atlas in-context aesthetic-grammar

FLUX.1 Kontext - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for FLUX.1 Kontext maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-198; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

FLUX.1 Kontext - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for FLUX.1 Kontext is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For FLUX.1 Kontext: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-199; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

FLUX.1 Kontext - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for FLUX.1 Kontext converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for FLUX.1 Kontext: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-200; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

FLUX.1 Kontext - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for FLUX.1 Kontext defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: iterative consistency is improved but not guaranteed; version every turn.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using FLUX.1 Kontext: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-201; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

model-atlas in-context evaluation-court

FLUX.1 Kontext - Evaluation Court

CLAIM / FUNCTION

Evaluation court for FLUX.1 Kontext uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of FLUX.1 Kontext on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-202; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

FLUX.1 Kontext - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for FLUX.1 Kontext must be known before deployment: iterative consistency is improved but not guaranteed; version every turn.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on FLUX.1 Kontext output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-203; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

model-atlas in-context rag/graphrag-metadata

FLUX.1 Kontext - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for FLUX.1 Kontext stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a FLUX.1 Kontext generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-204; model=FLUX.1 Kontext; confidence=Supported; update_status=watch; risk=iterative consistency is improved but not guaranteed; version every turn

model-atlas customizable positioning-&-ontology

Stable Diffusion 3.5 Large - Positioning & Ontology

CLAIM / FUNCTION

Stable Diffusion 3.5 Large is treated as a bounded production organ, not a universal answer. Type: customizable open image model. Best use: LoRA, ControlNet-style pipelines, privacy, local/fine-tuned brand systems. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Stable Diffusion 3.5 Large only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-205; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Commercial Use Map

CLAIM / FUNCTION

Commercially, Stable Diffusion 3.5 Large must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Stable Diffusion 3.5 Large, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Stable Diffusion 3.5 Large: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-206; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Stable Diffusion 3.5 Large maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-207; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Stable Diffusion 3.5 Large is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Stable Diffusion 3.5 Large: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-208; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Stable Diffusion 3.5 Large converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Stable Diffusion 3.5 Large: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-209; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Stable Diffusion 3.5 Large defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: license and commercial terms depend on revenue/use case.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Stable Diffusion 3.5 Large: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-210; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Stable Diffusion 3.5 Large uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Stable Diffusion 3.5 Large on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-211; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Stable Diffusion 3.5 Large must be known before deployment: license and commercial terms depend on revenue/use case.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Stable Diffusion 3.5 Large output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-212; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Large - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Stable Diffusion 3.5 Large stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Stable Diffusion 3.5 Large generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-213; model=Stable Diffusion 3.5 Large; confidence=Supported; update_status=watch; risk=license and commercial terms depend on revenue/use case

Stable Diffusion 3.5 Turbo/Flash - Positioning & Ontology

CLAIM / FUNCTION

Stable Diffusion 3.5 Turbo/Flash is treated as a bounded production organ, not a universal answer. Type: fast/efficient image generation. Best use: rapid prototypes, device-constrained experiments, cheap iteration. Epistemic rank: Supported/implementation dependent.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Stable Diffusion 3.5 Turbo/Flash only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-214; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - Commercial Use Map

CLAIM / FUNCTION

Commercially, Stable Diffusion 3.5 Turbo/Flash must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Stable Diffusion 3.5 Turbo/Flash, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Stable Diffusion 3.5 Turbo/Flash: 1 big title,
1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-215; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Stable Diffusion 3.5 Turbo/Flash maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-216; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Stable Diffusion 3.5 Turbo/Flash is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Stable Diffusion 3.5 Turbo/Flash: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-217; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Stable Diffusion 3.5 Turbo/Flash converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Stable Diffusion 3.5 Turbo/Flash: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-218; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Stable Diffusion 3.5 Turbo/Flash defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: quality/faithfulness trade-offs need measurement.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Stable Diffusion 3.5 Turbo/Flash:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-219; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

model-atlas fast/efficient evaluation-court

Stable Diffusion 3.5 Turbo/Flash - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Stable Diffusion 3.5 Turbo/Flash uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Stable Diffusion 3.5 Turbo/Flash on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-220; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

model-atlas fast/efficient failure-modes-&-repair

Stable Diffusion 3.5 Turbo/Flash - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Stable Diffusion 3.5 Turbo/Flash must be known before deployment: quality/faithfulness trade-offs need measurement.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Stable Diffusion 3.5 Turbo/Flash output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-221; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

Stable Diffusion 3.5 Turbo/Flash - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Stable Diffusion 3.5 Turbo/Flash stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Stable Diffusion 3.5 Turbo/Flash
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-222; model=Stable Diffusion 3.5 Turbo/Flash; confidence=Supported/implementation dependent; update_status=watch; risk=quality/faithfulness trade-offs need measurement

model-atlas design/vector/text-aware positioning-&-ontology

Recraft V3 - Positioning & Ontology

CLAIM / FUNCTION

Recraft V3 is treated as a bounded production organ, not a universal answer. Type: design/vector/text-aware image model. Best use: logos, icons, vector-oriented assets, text placement, ad graphics. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Recraft V3 only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-223; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas design/vector/text-aware commercial-use-map

Recraft V3 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Recraft V3 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Recraft V3, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Recraft V3: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-224; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas | design/vector/text-aware | aesthetic-grammar

Recraft V3 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Recraft V3 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-225; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

Recraft V3 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Recraft V3 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Recraft V3: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-226; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas design/vector/text-aware shot/motion-grammar

Recraft V3 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Recraft V3 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Recraft V3: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-227; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas | design/vector/text-aware | pipeline-&-toolchain

Recraft V3 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Recraft V3 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: final logos still require human vector/legal review.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Recraft V3: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-228; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

Recraft V3 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Recraft V3 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Recraft V3 on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-229; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas | design/vector/text-aware | failure-modes-&-repair

Recraft V3 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Recraft V3 must be known before deployment: final logos still require human vector/legal review.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Recraft V3 output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-230; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

model-atlas | design/vector/text-aware | rag/graphrag-metadata

Recraft V3 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Recraft V3 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Recraft V3 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-231; model=Recraft V3; confidence=Supported; update_status=watch; risk=final logos still require human vector/legal review

Leonardo Phoenix - Positioning & Ontology

CLAIM / FUNCTION

Leonardo Phoenix is treated as a bounded production organ, not a universal answer. Type: prompt-control image model. Best use: precise prompt adherence, clear text, high-resolution brand imagery. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Leonardo Phoenix only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-232; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Commercial Use Map

CLAIM / FUNCTION

Commercially, Leonardo Phoenix must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Leonardo Phoenix, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Leonardo Phoenix: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-233; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Leonardo Phoenix maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-234; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Leonardo Phoenix is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Leonardo Phoenix: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-235; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Leonardo Phoenix converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Leonardo Phoenix: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-236; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Leonardo Phoenix defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: API settings and ultra/quality modes affect outcomes.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Leonardo Phoenix: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-237; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Leonardo Phoenix uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Leonardo Phoenix on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-238; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Leonardo Phoenix must be known before deployment: API settings and ultra/quality modes affect outcomes.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Leonardo Phoenix output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-239; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

Leonardo Phoenix - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Leonardo Phoenix stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Leonardo Phoenix generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-240; model=Leonardo Phoenix; confidence=Supported; update_status=watch; risk=API settings and ultra/quality modes affect outcomes

model-atlas controllable positioning-&-ontology

Luma Ray 3.2 - Positioning & Ontology

CLAIM / FUNCTION

Luma Ray 3.2 is treated as a bounded production organ, not a universal answer. Type: controllable video generation. Best use: frame-directed cinematic cuts, continuity workflows, scalable video direction. Epistemic rank: Supported by official page.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Luma Ray 3.2 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-241; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Luma Ray 3.2 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Luma Ray 3.2, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Luma Ray 3.2: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-242; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Luma Ray 3.2 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-243; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Luma Ray 3.2 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Luma Ray 3.2: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-244; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Luma Ray 3.2 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Luma Ray 3.2: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-245; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

model-atlas controllable pipeline-&-toolchain

Luma Ray 3.2 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Luma Ray 3.2 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: official feature availability must be confirmed per plan/API.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Luma Ray 3.2: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-246; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Luma Ray 3.2 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Luma Ray 3.2 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-247; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Luma Ray 3.2 must be known before deployment: official feature availability must be confirmed per plan/API.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Luma Ray 3.2 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-248; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

Luma Ray 3.2 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Luma Ray 3.2 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Luma Ray 3.2 generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-249; model=Luma Ray 3.2; confidence=Supported by official page; update_status=watch; risk=official feature availability must be confirmed per plan/API

model-atlas realistic positioning-&-ontology

Luma Ray 2 - Positioning & Ontology

CLAIM / FUNCTION

Luma Ray 2 is treated as a bounded production organ, not a universal answer. Type: realistic motion video model. Best use: natural motion, cinematic b-roll, image-to-video and physics-sensitive shots. Epistemic rank: Supported/plausible.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Luma Ray 2 only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-250; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

Luma Ray 2 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Luma Ray 2 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Luma Ray 2, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Luma Ray 2: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-251; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

Luma Ray 2 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Luma Ray 2 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-252; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

Luma Ray 2 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Luma Ray 2 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Luma Ray 2: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-253; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

Luma Ray 2 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Luma Ray 2 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Luma Ray 2: 0-2s hook, 2-5s surreal proof, 5-8s product
clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with
motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-254; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

Luma Ray 2 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Luma Ray 2 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: model/version may differ across integrators.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Luma Ray 2: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-255; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

model-atlas realistic evaluation-court

Luma Ray 2 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Luma Ray 2 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Luma Ray 2 on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-256; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

model-atlas realistic failure-modes-&-repair

Luma Ray 2 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Luma Ray 2 must be known before deployment: model/version may differ across integrators.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Luma Ray 2 output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-257; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

model-atlas realistic rag/graphrag-metadata

Luma Ray 2 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Luma Ray 2 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Luma Ray 2 generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-258; model=Luma Ray 2; confidence=Supported/plausible; update_status=watch; risk=model/version may differ across integrators

model-atlas video/image/sound positioning-&-ontology

Kling 3.0 Video / Omni - Positioning & Ontology

CLAIM / FUNCTION

Kling 3.0 Video / Omni is treated as a bounded production organ, not a universal answer. Type: video/image/sound creative studio. Best use: native 4K claims, multimodal instruction parsing, light/sound narrative tests. Epistemic rank: Supported by official page with claim-check.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Kling 3.0 Video / Omni only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-259; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

Kling 3.0 Video / Omni - Commercial Use Map

CLAIM / FUNCTION

Commercially, Kling 3.0 Video / Omni must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Kling 3.0 Video / Omni, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Kling 3.0 Video / Omni: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-260; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

model-atlas video/image/sound aesthetic-grammar

Kling 3.0 Video / Omni - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Kling 3.0 Video / Omni maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-261; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

model-atlas video/image/sound prompt-injection-loop

Kling 3.0 Video / Omni - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Kling 3.0 Video / Omni is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Kling 3.0 Video / Omni: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-262; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

Kling 3.0 Video / Omni - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Kling 3.0 Video / Omni converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Kling 3.0 Video / Omni: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-263; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

Kling 3.0 Video / Omni - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Kling 3.0 Video / Omni defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: rights, region, API terms and model claims require verification.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Kling 3.0 Video / Omni: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-264; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

Kling 3.0 Video / Omni - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Kling 3.0 Video / Omni uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Kling 3.0 Video / Omni on 10 criteria. For each
weak point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-265; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

Kling 3.0 Video / Omni - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Kling 3.0 Video / Omni must be known before deployment: rights, region, API terms and model claims require verification.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Kling 3.0 Video / Omni output: generic style,
wrong text, object drift, rights risk, uncanny motion, overclaim, weak
hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-266; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

model-atlas video/image/sound rag/graphrag-metadata

Kling 3.0 Video / Omni - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Kling 3.0 Video / Omni stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Kling 3.0 Video / Omni generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-267; model=Kling 3.0 Video / Omni; confidence=Supported by official page with claim-check; update_status=watch; risk=rights, region, API terms and model claims require verification

MiniMax Hailuo 02 - Positioning & Ontology

CLAIM / FUNCTION

MiniMax Hailuo 02 is treated as a bounded production organ, not a universal answer. Type: short-form video generation. Best use: physics-heavy social clips, gymnastic/action tests, 1080p experiments. Epistemic rank: Supported with legal caution.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to MiniMax Hailuo 02 only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-268; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Commercial Use Map

CLAIM / FUNCTION

Commercially, MiniMax Hailuo 02 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For MiniMax Hailuo 02, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for MiniMax Hailuo 02: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-269; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for MiniMax Hailuo 02 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-270; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

model-atlas short-form prompt-injection-loop

MiniMax Hailuo 02 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for MiniMax Hailuo 02 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For MiniMax Hailuo 02: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-271; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for MiniMax Hailuo 02 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for MiniMax Hailuo 02: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-272; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for MiniMax Hailuo 02 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: copyright litigation context and IP safeguards require strict governance.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using MiniMax Hailuo 02: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-273; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for MiniMax Hailuo 02 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of MiniMax Hailuo 02 on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-274; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for MiniMax Hailuo 02 must be known before deployment: copyright litigation context and IP safeguards require strict governance.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on MiniMax Hailuo 02 output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-275; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

MiniMax Hailuo 02 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for MiniMax Hailuo 02 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a MiniMax Hailuo 02 generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-276; model=MiniMax Hailuo 02; confidence=Supported with legal caution; update_status=watch; risk=copyright litigation context and IP safeguards require strict governance

model-atlas social positioning-&-ontology

Pika / Trendmaker - Positioning & Ontology

CLAIM / FUNCTION

Pika / Trendmaker is treated as a bounded production organ, not a universal answer. Type: social video and effects. Best use: viral trend formats, selfie-to-scene, stylized clips, performance hooks. Epistemic rank: Supported by product page.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Pika / Trendmaker only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-277; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Commercial Use Map

CLAIM / FUNCTION

Commercially, Pika / Trendmaker must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Pika / Trendmaker, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Pika / Trendmaker: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-278; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Pika / Trendmaker maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-279; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Pika / Trendmaker is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Pika / Trendmaker: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-280; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Pika / Trendmaker converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Pika / Trendmaker: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-281; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Pika / Trendmaker defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: avoid fake endorsements, deceptive UGC and likeness misuse.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Pika / Trendmaker: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-282; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Pika / Trendmaker uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Pika / Trendmaker on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-283; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

Pika / Trendmaker - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Pika / Trendmaker must be known before deployment: avoid fake endorsements, deceptive UGC and likeness misuse.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Pika / Trendmaker output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-284; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

model-atlas social rag/graphrag-metadata

Pika / Trendmaker - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Pika / Trendmaker stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Pika / Trendmaker generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-285; model=Pika / Trendmaker; confidence=Supported by product page; update_status=watch; risk=avoid fake endorsements, deceptive UGC and likeness misuse

PixVerse - Positioning & Ontology

CLAIM / FUNCTION

PixVerse is treated as a bounded production organ, not a universal answer. Type: social video generator. Best use: quick vertical ad variants, template-style motion and trend testing. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to PixVerse only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-286; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

PixVerse - Commercial Use Map

CLAIM / FUNCTION

Commercially, PixVerse must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For PixVerse, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for PixVerse: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-287; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

model-atlas social aesthetic-grammar

PixVerse - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for PixVerse maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-288; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

model-atlas social prompt-injection-loop

PixVerse - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for PixVerse is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For PixVerse: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-289; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

PixVerse - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for PixVerse converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for PixVerse: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-290; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

PixVerse - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for PixVerse defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: verify model version and commercial rights before client delivery.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using PixVerse: inputs, model settings,
reference files, post tools, QA gates, export specs, metadata, rollback
plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-291; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

PixVerse - Evaluation Court

CLAIM / FUNCTION

Evaluation court for PixVerse uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of PixVerse on 10 criteria. For each weak point,
write a repair instruction and decide: regenerate, edit, composite, or
reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-292; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

PixVerse - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for PixVerse must be known before deployment: verify model version and commercial rights before client delivery.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on PixVerse output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-293; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

model-atlas social rag/graphrag-metadata

PixVerse - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for PixVerse stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a PixVerse generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-294; model=PixVerse; confidence=Needs verification; update_status=watch; risk=verify model version and commercial rights before client delivery

model-atlas | open/Chinese | positioning-&-ontology

Wan 2.x video slot - Positioning & Ontology

CLAIM / FUNCTION

Wan 2.x video slot is treated as a bounded production organ, not a universal answer. Type: open/Chinese video ecosystem. Best use: local video experiments, research comparisons, model-routing alternatives. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Wan 2.x video slot only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-295; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

Wan 2.x video slot - Commercial Use Map

CLAIM / FUNCTION

Commercially, Wan 2.x video slot must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Wan 2.x video slot, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Wan 2.x video slot: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-296; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas open/Chinese aesthetic-grammar

Wan 2.x video slot - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Wan 2.x video slot maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-297; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas open/Chinese prompt-injection-loop

Wan 2.x video slot - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Wan 2.x video slot is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Wan 2.x video slot: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-298; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas | open/Chinese | shot/motion-grammar

Wan 2.x video slot - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Wan 2.x video slot converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Wan 2.x video slot: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-299; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas | open/Chinese | pipeline-&-toolchain

Wan 2.x video slot - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Wan 2.x video slot defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: license, safety model card and compute must be checked.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Wan 2.x video slot: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-300; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas open/Chinese evaluation-court

Wan 2.x video slot - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Wan 2.x video slot uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Wan 2.x video slot on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-301; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

Wan 2.x video slot - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Wan 2.x video slot must be known before deployment: license, safety model card and compute must be checked.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Wan 2.x video slot output: generic style,
wrong text, object drift, rights risk, uncanny motion, overclaim, weak
hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-302; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas open/Chinese rag/graphrag-metadata

Wan 2.x video slot - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Wan 2.x video slot stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Wan 2.x video slot generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-303; model=Wan 2.x video slot; confidence=Needs verification; update_status=watch; risk=license, safety model card and compute must be checked

model-atlas open positioning-&-ontology

HunyuanVideo - Positioning & Ontology

CLAIM / FUNCTION

HunyuanVideo is treated as a bounded production organ, not a universal answer. Type: open video model. Best use: local/research comparison, video generation stack testing. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to HunyuanVideo only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-304; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

model-atlas open commercial-use-map

HunyuanVideo - Commercial Use Map

CLAIM / FUNCTION

Commercially, HunyuanVideo must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For HunyuanVideo, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for HunyuanVideo: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-305; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

HunyuanVideo - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for HunyuanVideo maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-306; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

model-atlas open prompt-injection-loop

HunyuanVideo - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for HunyuanVideo is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For HunyuanVideo: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-307; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

HunyuanVideo - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for HunyuanVideo converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for HunyuanVideo: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-308; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

HunyuanVideo - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for HunyuanVideo defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: compute-heavy, not automatically client-ready.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using HunyuanVideo: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-309; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

HunyuanVideo - Evaluation Court

CLAIM / FUNCTION

Evaluation court for HunyuanVideo uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of HunyuanVideo on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-310; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

HunyuanVideo - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for HunyuanVideo must be known before deployment: compute-heavy, not automatically client-ready.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on HunyuanVideo output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-311; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

model-atlas open rag/graphrag-metadata

HunyuanVideo - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for HunyuanVideo stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a HunyuanVideo generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-312; model=HunyuanVideo; confidence=Needs verification; update_status=watch; risk=compute-heavy, not automatically client-ready

model-atlas open-source positioning-&-ontology

Open-Sora 2.0 - Positioning & Ontology

CLAIM / FUNCTION

Open-Sora 2.0 is treated as a bounded production organ, not a universal answer. Type: open-source commercial-level video research. Best use: learning video training/optimization techniques, benchmarks, open pipelines. Epistemic rank: Supported by paper.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Open-Sora 2.0 only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-313; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - Commercial Use Map

CLAIM / FUNCTION

Commercially, Open-Sora 2.0 must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Open-Sora 2.0, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Open-Sora 2.0: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-314; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Open-Sora 2.0 maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-315; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

model-atlas open-source prompt-injection-loop

Open-Sora 2.0 - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Open-Sora 2.0 is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Open-Sora 2.0: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-316; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

model-atlas open-source shot/motion-grammar

Open-Sora 2.0 - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Open-Sora 2.0 converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Open-Sora 2.0: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-317; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Open-Sora 2.0 defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: do not confuse open research reproduction with turnkey agency output.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Open-Sora 2.0: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-318; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Open-Sora 2.0 uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Open-Sora 2.0 on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-319; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Open-Sora 2.0 must be known before deployment: do not confuse open research reproduction with turnkey agency output.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Open-Sora 2.0 output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-320; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

Open-Sora 2.0 - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Open-Sora 2.0 stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Open-Sora 2.0 generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-321; model=Open-Sora 2.0; confidence=Supported by paper; update_status=watch; risk=do not confuse open research reproduction with turnkey agency output

model-atlas open positioning-&-ontology

CogVideoX - Positioning & Ontology

CLAIM / FUNCTION

CogVideoX is treated as a bounded production organ, not a universal answer. Type: open video model. Best use: local video experiments, code-driven research and comparative evaluation. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to CogVideoX only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-322; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Commercial Use Map

CLAIM / FUNCTION

Commercially, CogVideoX must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For CogVideoX, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for CogVideoX: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-323; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for CogVideoX maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-324; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for CogVideoX is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For CogVideoX: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-325; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for CogVideoX converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for CogVideoX: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-326; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for CogVideoX defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: quality and commercial terms must be validated.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using CogVideoX: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-327; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - Evaluation Court

CLAIM / FUNCTION

Evaluation court for CogVideoX uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of CogVideoX on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-328; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

model-atlas open failure-modes-&-repair

CogVideoX - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for CogVideoX must be known before deployment: quality and commercial terms must be validated.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on CogVideoX output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-329; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

CogVideoX - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for CogVideoX stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a CogVideoX generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-330; model=CogVideoX; confidence=Needs verification; update_status=watch; risk=quality and commercial terms must be validated

model-atlas open/creative positioning-&-ontology

Mochi / Genmo - Positioning & Ontology

CLAIM / FUNCTION

Mochi / Genmo is treated as a bounded production organ, not a universal answer. Type: open/creative video model. Best use: motion studies, open video pipelines and experimental clips. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Mochi / Genmo only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-331; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

Mochi / Genmo - Commercial Use Map

CLAIM / FUNCTION

Commercially, Mochi / Genmo must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Mochi / Genmo, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Mochi / Genmo: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-332; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

Mochi / Genmo - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Mochi / Genmo maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-333; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas open/creative prompt-injection-loop

Mochi / Genmo - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Mochi / Genmo is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Mochi / Genmo: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-334; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas open/creative shot/motion-grammar

Mochi / Genmo - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Mochi / Genmo converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Mochi / Genmo: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-335; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

Mochi / Genmo - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Mochi / Genmo defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: may need strong finishing and evaluation.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Mochi / Genmo: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-336; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas open/creative evaluation-court

Mochi / Genmo - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Mochi / Genmo uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Mochi / Genmo on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-337; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas open/creative failure-modes-&-repair

Mochi / Genmo - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Mochi / Genmo must be known before deployment: may need strong finishing and evaluation.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Mochi / Genmo output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-338; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas open/creative rag/graphrag-metadata

Mochi / Genmo - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Mochi / Genmo stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Mochi / Genmo generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-339; model=Mochi / Genmo; confidence=Needs verification; update_status=watch; risk=may need strong finishing and evaluation

model-atlas animation positioning-&-ontology

AnimateDiff / Deforum - Positioning & Ontology

CLAIM / FUNCTION

AnimateDiff / Deforum is treated as a bounded production organ, not a universal answer. Type: animation add-ons. Best use: music-video, psychedelic, style-continuity and motion-poster experiments. Epistemic rank: Established tools.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to AnimateDiff / Deforum only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-340; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

model-atlas animation commercial-use-map

AnimateDiff / Deforum - Commercial Use Map

CLAIM / FUNCTION

Commercially, AnimateDiff / Deforum must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For AnimateDiff / Deforum, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for AnimateDiff / Deforum: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-341; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

model-atlas animation aesthetic-grammar

AnimateDiff / Deforum - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for AnimateDiff / Deforum maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-342; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

model-atlas animation prompt-injection-loop

AnimateDiff / Deforum - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for AnimateDiff / Deforum is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For AnimateDiff / Deforum: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-343; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

model-atlas animation shot/motion-grammar

AnimateDiff / Deforum - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for AnimateDiff / Deforum converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for AnimateDiff / Deforum: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-344; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

AnimateDiff / Deforum - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for AnimateDiff / Deforum defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: temporal artifacts and prompt drift are common.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using AnimateDiff / Deforum: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-345; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

model-atlas animation evaluation-court

AnimateDiff / Deforum - Evaluation Court

CLAIM / FUNCTION

Evaluation court for AnimateDiff / Deforum uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of AnimateDiff / Deforum on 10 criteria. For each
weak point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-346; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

AnimateDiff / Deforum - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for AnimateDiff / Deforum must be known before deployment: temporal artifacts and prompt drift are common.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on AnimateDiff / Deforum output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-347; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

AnimateDiff / Deforum - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for AnimateDiff / Deforum stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a AnimateDiff / Deforum generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-348; model=AnimateDiff / Deforum; confidence=Established tools; update_status=watch; risk=temporal artifacts and prompt drift are common

ComfyUI - Positioning & Ontology

CLAIM / FUNCTION

ComfyUI is treated as a bounded production organ, not a universal answer. Type: node-based generative pipeline. Best use: reproducible ControlNet/IP-Adapter/LoRA/mask pipelines. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :
.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to ComfyUI only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-349; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

ComfyUI - Commercial Use Map

CLAIM / FUNCTION

Commercially, ComfyUI must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For ComfyUI, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for ComfyUI: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-350; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

ComfyUI - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for ComfyUI maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-351; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

ComfyUI - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for ComfyUI is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For ComfyUI: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-352; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

model-atlas node-based shot/motion-grammar

ComfyUI - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for ComfyUI converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for ComfyUI: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-353; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

ComfyUI - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for ComfyUI defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: unversioned graphs become unmaintainable spaghetti.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using ComfyUI: inputs, model settings,
reference files, post tools, QA gates, export specs, metadata, rollback
plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-354; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

ComfyUI - Evaluation Court

CLAIM / FUNCTION

Evaluation court for ComfyUI uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of ComfyUI on 10 criteria. For each weak point,
write a repair instruction and decide: regenerate, edit, composite, or
reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-355; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

model-atlas node-based failure-modes-&-repair

ComfyUI - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for ComfyUI must be known before deployment: unversioned graphs become unmaintainable spaghetti.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on ComfyUI output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-356; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

model-atlas node-based rag/graphrag-metadata

ComfyUI - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for ComfyUI stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a ComfyUI generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-357; model=ComfyUI; confidence=Established; update_status=watch; risk=unversioned graphs become unmaintainable spaghetti

model-atlas | developer | positioning-&-ontology

Hugging Face Diffusers - Positioning & Ontology

CLAIM / FUNCTION

Hugging Face Diffusers is treated as a bounded production organ, not a universal answer. Type: developer inference/fine-tuning pipeline. Best use: reproducible research, evaluations, custom scripts, local deployment. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Hugging Face Diffusers only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-358; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Commercial Use Map

CLAIM / FUNCTION

Commercially, Hugging Face Diffusers must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Hugging Face Diffusers, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Hugging Face Diffusers: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-359; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Hugging Face Diffusers maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-360; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Hugging Face Diffusers is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Hugging Face Diffusers: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-361; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

model-atlas | developer | shot/motion-grammar

Hugging Face Diffusers - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Hugging Face Diffusers converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Hugging Face Diffusers: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-362; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Hugging Face Diffusers defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: requires engineering discipline and license checks.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Hugging Face Diffusers: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-363; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Hugging Face Diffusers uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Hugging Face Diffusers on 10 criteria. For each
weak point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-364; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

Hugging Face Diffusers - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Hugging Face Diffusers must be known before deployment: requires engineering discipline and license checks.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Hugging Face Diffusers output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-365; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

model-atlas | developer | rag/graphrag-metadata

Hugging Face Diffusers - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Hugging Face Diffusers stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Hugging Face Diffusers generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-366; model=Hugging Face Diffusers; confidence=Established; update_status=watch; risk=requires engineering discipline and license checks

ControlNet family - Positioning & Ontology

CLAIM / FUNCTION

ControlNet family is treated as a bounded production organ, not a universal answer. Type: conditioning control. Best use: pose, depth, edge, canny, lineart, layout-preserving generation. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to ControlNet family only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-367; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - Commercial Use Map

CLAIM / FUNCTION

Commercially, ControlNet family must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For ControlNet family, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for ControlNet family: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-368; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for ControlNet family maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-369; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

model-atlas conditioning prompt-injection-loop

ControlNet family - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for ControlNet family is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For ControlNet family: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-370; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for ControlNet family converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for ControlNet family: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-371; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

model-atlas conditioning pipeline-&-toolchain

ControlNet family - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for ControlNet family defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: overconstraint can produce rigid or uncanny results.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using ControlNet family: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-372; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - Evaluation Court

CLAIM / FUNCTION

Evaluation court for ControlNet family uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of ControlNet family on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-373; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for ControlNet family must be known before deployment: overconstraint can produce rigid or uncanny results.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on ControlNet family output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-374; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

ControlNet family - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for ControlNet family stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a ControlNet family generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-375; model=ControlNet family; confidence=Established; update_status=watch; risk=overconstraint can produce rigid or uncanny results

IP-Adapter / reference adapters - Positioning & Ontology

CLAIM / FUNCTION

IP-Adapter / reference adapters is treated as a bounded production organ, not a universal answer. Type: style and identity reference control. Best use: brand tokens, product families, character consistency, style memory. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to IP-Adapter / reference adapters only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-376; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - Commercial Use Map

CLAIM / FUNCTION

Commercially, IP-Adapter / reference adapters must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For IP-Adapter / reference adapters, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for IP-Adapter / reference adapters: 1 big title,
1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-377; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for IP-Adapter / reference adapters maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-378; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

model-atlas style prompt-injection-loop

IP-Adapter / reference adapters - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for IP-Adapter / reference adapters is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For IP-Adapter / reference adapters: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-379; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for IP-Adapter / reference adapters converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for IP-Adapter / reference adapters: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-380; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for IP-Adapter / reference adapters defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: only use owned/licensed references; check resemblance risks.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using IP-Adapter / reference adapters:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-381; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

model-atlas style evaluation-court

IP-Adapter / reference adapters - Evaluation Court

CLAIM / FUNCTION

Evaluation court for IP-Adapter / reference adapters uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of IP-Adapter / reference adapters on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-382; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for IP-Adapter / reference adapters must be known before deployment: only use owned/licensed references; check resemblance risks.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on IP-Adapter / reference adapters output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-383; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

IP-Adapter / reference adapters - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for IP-Adapter / reference adapters stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a IP-Adapter / reference adapters
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-384; model=IP-Adapter / reference adapters; confidence=Established; update_status=watch; risk=only use owned/licensed references; check resemblance risks

model-atlas fine-tuned positioning-&-ontology

LoRA / DreamBooth - Positioning & Ontology

CLAIM / FUNCTION

LoRA / DreamBooth is treated as a bounded production organ, not a universal answer. Type: fine-tuned visual memory. Best use: brand/product/character consistency when prompts alone fail. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to LoRA / DreamBooth only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-385; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

LoRA / DreamBooth - Commercial Use Map

CLAIM / FUNCTION

Commercially, LoRA / DreamBooth must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For LoRA / DreamBooth, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for LoRA / DreamBooth: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-386; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

LoRA / DreamBooth - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for LoRA / DreamBooth maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-387; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

model-atlas fine-tuned prompt-injection-loop

LoRA / DreamBooth - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for LoRA / DreamBooth is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For LoRA / DreamBooth: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-388; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

LoRA / DreamBooth - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for LoRA / DreamBooth converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for LoRA / DreamBooth: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-389; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

LoRA / DreamBooth - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for LoRA / DreamBooth defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: overfitting, leakage and rights issues.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using LoRA / DreamBooth: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-390; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

model-atlas fine-tuned evaluation-court

LoRA / DreamBooth - Evaluation Court

CLAIM / FUNCTION

Evaluation court for LoRA / DreamBooth uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of LoRA / DreamBooth on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-391; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

LoRA / DreamBooth - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for LoRA / DreamBooth must be known before deployment: overfitting, leakage and rights issues.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on LoRA / DreamBooth output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-392; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

model-atlas fine-tuned rag/graphrag-metadata

LoRA / DreamBooth - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for LoRA / DreamBooth stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a LoRA / DreamBooth generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-393; model=LoRA / DreamBooth; confidence=Established; update_status=watch; risk=overfitting, leakage and rights issues

Segment Anything / masking - Positioning & Ontology

CLAIM / FUNCTION

Segment Anything / masking is treated as a bounded production organ, not a universal answer. Type: segmentation and edit preparation. Best use: product isolation, precise background swaps, ad retouching pipelines. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Segment Anything / masking only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-394; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - Commercial Use Map

CLAIM / FUNCTION

Commercially, Segment Anything / masking must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Segment Anything / masking, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Segment Anything / masking: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-395; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Segment Anything / masking maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-396; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Segment Anything / masking is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Segment Anything / masking: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-397; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Segment Anything / masking converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Segment Anything / masking: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-398; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Segment Anything / masking defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: mask edges and transparency must be reviewed.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Segment Anything / masking: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-399; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

model-atlas segmentation evaluation-court

Segment Anything / masking - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Segment Anything / masking uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Segment Anything / masking on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-400; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

model-atlas segmentation failure-modes-&-repair

Segment Anything / masking - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Segment Anything / masking must be known before deployment: mask edges and transparency must be reviewed.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Segment Anything / masking output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-401; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

Segment Anything / masking - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Segment Anything / masking stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Segment Anything / masking generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-402; model=Segment Anything / masking; confidence=Established; update_status=watch; risk=mask edges and transparency must be reviewed

model-atlas structural positioning-&-ontology

Depth/Pose/Lineart preprocessors - Positioning & Ontology

CLAIM / FUNCTION

Depth/Pose/Lineart preprocessors is treated as a bounded production organ, not a universal answer. Type: structural control. Best use: storyboards, anatomy, perspective, architecture, product geometry. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Depth/Pose/Lineart preprocessors only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-403; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural commercial-use-map

Depth/Pose/Lineart preprocessors - Commercial Use Map

CLAIM / FUNCTION

Commercially, Depth/Pose/Lineart preprocessors must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Depth/Pose/Lineart preprocessors, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Depth/Pose/Lineart preprocessors: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-404; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural aesthetic-grammar

Depth/Pose/Lineart preprocessors - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Depth/Pose/Lineart preprocessors maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-405; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural prompt-injection-loop

Depth/Pose/Lineart preprocessors - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Depth/Pose/Lineart preprocessors is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Depth/Pose/Lineart preprocessors: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-406; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural shot/motion-grammar

Depth/Pose/Lineart preprocessors - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Depth/Pose/Lineart preprocessors converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Depth/Pose/Lineart preprocessors: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-407; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural pipeline-&-toolchain

Depth/Pose/Lineart preprocessors - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Depth/Pose/Lineart preprocessors defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: bad control image creates bad generation.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Depth/Pose/Lineart preprocessors:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-408; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural evaluation-court

Depth/Pose/Lineart preprocessors - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Depth/Pose/Lineart preprocessors uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Depth/Pose/Lineart preprocessors on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-409; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural failure-modes-&-repair

Depth/Pose/Lineart preprocessors - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Depth/Pose/Lineart preprocessors must be known before deployment: bad control image creates bad generation.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Depth/Pose/Lineart preprocessors output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-410; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas structural rag/graphrag-metadata

Depth/Pose/Lineart preprocessors - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Depth/Pose/Lineart preprocessors stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Depth/Pose/Lineart preprocessors
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-411; model=Depth/Pose/Lineart preprocessors; confidence=Established; update_status=watch; risk=bad control image creates bad generation

model-atlas | upscale/denoise/interpolatio | positioning-&-ontology

Topaz Video AI - Positioning & Ontology

CLAIM / FUNCTION

Topaz Video AI is treated as a bounded production organ, not a universal answer. Type: upscale/denoise/interpolation. Best use: finishing low-res/generated clips, motion repair, sharpening. Epistemic rank: Established.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Topaz Video AI only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-412; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio commercial-use-map

Topaz Video AI - Commercial Use Map

CLAIM / FUNCTION

Commercially, Topaz Video AI must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Topaz Video AI, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Topaz Video AI: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-413; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio aesthetic-grammar

Topaz Video AI - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Topaz Video AI maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-414; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio prompt-injection-loop

Topaz Video AI - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Topaz Video AI is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNE

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
[HYPERPROMPT] For Topaz Video AI: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-415; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio shot/motion-grammar

Topaz Video AI - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Topaz Video AI converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Topaz Video AI: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-416; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

Topaz Video AI - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Topaz Video AI defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: interpolation can invent artifacts.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Topaz Video AI: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-417; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio evaluation-court

Topaz Video AI - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Topaz Video AI uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Topaz Video AI on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-418; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas upscale/denoise/interpolatio failure-modes-&-repair

Topaz Video AI - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Topaz Video AI must be known before deployment: interpolation can invent artifacts.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Topaz Video AI output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-419; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

Topaz Video AI - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Topaz Video AI stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a Topaz Video AI generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-420; model=Topaz Video AI; confidence=Established; update_status=watch; risk=interpolation can invent artifacts

model-atlas detail positioning-&-ontology

Magnific / creative upscalers - Positioning & Ontology

CLAIM / FUNCTION

Magnific / creative upscalers is treated as a bounded production organ, not a universal answer. Type: detail hallucination and upscaling. Best use: hero image texture, expressive detail, artistic enhancement. Epistemic rank: Needs verification.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Magnific / creative upscalers only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-421; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

model-atlas detail commercial-use-map

Magnific / creative upscalers - Commercial Use Map

CLAIM / FUNCTION

Commercially, Magnific / creative upscalers must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Magnific / creative upscalers, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Magnific / creative upscalers: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-422; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

model-atlas detail aesthetic-grammar

Magnific / creative upscalers - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Magnific / creative upscalers maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-423; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Magnific / creative upscalers is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Magnific / creative upscalers: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-424; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Magnific / creative upscalers converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Magnific / creative upscalers: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-425; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Magnific / creative upscalers defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: can hallucinate product details and legal-sensitive features.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Magnific / creative upscalers: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-426; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Magnific / creative upscalers uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Magnific / creative upscalers on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-427; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Magnific / creative upscalers must be known before deployment: can hallucinate product details and legal-sensitive features.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Magnific / creative upscalers output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-428; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Magnific / creative upscalers - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Magnific / creative upscalers stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a Magnific / creative upscalers
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-429; model=Magnific / creative upscalers; confidence=Needs verification; update_status=watch; risk=can hallucinate product details and legal-sensitive features

Krea real-time canvas - Positioning & Ontology

CLAIM / FUNCTION

Krea real-time canvas is treated as a bounded production organ, not a universal answer. Type: live ideation and creative enhancement. Best use: fast style exploration, visual direction sessions, real-time variants. Epistemic rank: Needs verification.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Krea real-time canvas only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-430; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Commercial Use Map

CLAIM / FUNCTION

Commercially, Krea real-time canvas must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Krea real-time canvas, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Krea real-time canvas: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-431; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Krea real-time canvas maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-432; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Krea real-time canvas is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Krea real-time canvas: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-433; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Krea real-time canvas converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Krea real-time canvas: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-434; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Krea real-time canvas defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: version and rights vary; do not confuse ideation with final proof.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Krea real-time canvas: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-435; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Krea real-time canvas uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of Krea real-time canvas on 10 criteria. For each
weak point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-436; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Krea real-time canvas - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Krea real-time canvas must be known before deployment: version and rights vary; do not confuse ideation with final proof.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Krea real-time canvas output: generic style,
wrong text, object drift, rights risk, uncanny motion, overclaim, weak
hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-437; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

model-atlas live rag/graphrag-metadata

Krea real-time canvas - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Krea real-time canvas stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Krea real-time canvas generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-438; model=Krea real-time canvas; confidence=Needs verification; update_status=watch; risk=version and rights vary; do not confuse ideation with final proof

Canva Magic Studio - Positioning & Ontology

CLAIM / FUNCTION

Canva Magic Studio is treated as a bounded production organ, not a universal answer. Type: business design layer. Best use: templates, social posts, non-designer production, fast adaptation. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Canva Magic Studio only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-439; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

Canva Magic Studio - Commercial Use Map

CLAIM / FUNCTION

Commercially, Canva Magic Studio must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Canva Magic Studio, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Canva Magic Studio: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-440; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

model-atlas business aesthetic-grammar

Canva Magic Studio - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Canva Magic Studio maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-441; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

Canva Magic Studio - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Canva Magic Studio is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Canva Magic Studio: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-442; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

Canva Magic Studio - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Canva Magic Studio converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Canva Magic Studio: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-443; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

Canva Magic Studio - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Canva Magic Studio defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: generic templates; enforce brand tokens and originality.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Canva Magic Studio: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-444; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

model-atlas business evaluation-court

Canva Magic Studio - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Canva Magic Studio uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Canva Magic Studio on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-445; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

Canva Magic Studio - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Canva Magic Studio must be known before deployment: generic templates; enforce brand tokens and originality.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Canva Magic Studio output: generic style,
wrong text, object drift, rights risk, uncanny motion, overclaim, weak
hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-446; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

model-atlas business rag/graphrag-metadata

Canva Magic Studio - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Canva Magic Studio stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Canva Magic Studio generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-447; model=Canva Magic Studio; confidence=Established; update_status=watch; risk=generic templates; enforce brand tokens and originality

model-atlas design-system positioning-&-ontology

Figma AI / FigJam - Positioning & Ontology

CLAIM / FUNCTION

Figma AI / FigJam is treated as a bounded production organ, not a universal answer. Type: design-system collaboration. Best use: boards, UI/creative systems, componentized visual planning. Epistemic rank: Established.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Figma AI / FigJam only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-448; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Commercial Use Map

CLAIM / FUNCTION

Commercially, Figma AI / FigJam must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Figma AI / FigJam, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Figma AI / FigJam: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-449; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

model-atlas design-system aesthetic-grammar

Figma AI / FigJam - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Figma AI / FigJam maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-450; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Figma AI / FigJam is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Figma AI / FigJam: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-451; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Figma AI / FigJam converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Figma AI / FigJam: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-452; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Figma AI / FigJam defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: AI output must be integrated into real design systems.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Figma AI / FigJam: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-453; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Figma AI / FigJam uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Figma AI / FigJam on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-454; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

Figma AI / FigJam - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Figma AI / FigJam must be known before deployment: AI output must be integrated into real design systems.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Figma AI / FigJam output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-455; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

model-atlas design-system rag/graphrag-metadata

Figma AI / FigJam - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Figma AI / FigJam stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Figma AI / FigJam generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-456; model=Figma AI / FigJam; confidence=Established; update_status=watch; risk=AI output must be integrated into real design systems

model-atlas image positioning-&-ontology

Photoshop Generative Fill - Positioning & Ontology

CLAIM / FUNCTION

Photoshop Generative Fill is treated as a bounded production organ, not a universal answer. Type: image edit/composite. Best use: repair, outpaint, product scene extension, background replacement. Epistemic rank: Established.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Photoshop Generative Fill only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-457; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas image commercial-use-map

Photoshop Generative Fill - Commercial Use Map

CLAIM / FUNCTION

Commercially, Photoshop Generative Fill must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Photoshop Generative Fill, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Photoshop Generative Fill: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-458; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas image aesthetic-grammar

Photoshop Generative Fill - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Photoshop Generative Fill maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-459; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas image prompt-injection-loop

Photoshop Generative Fill - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Photoshop Generative Fill is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Photoshop Generative Fill: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-460; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas | image | shot/motion-grammar

Photoshop Generative Fill - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Photoshop Generative Fill converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Photoshop Generative Fill: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-461; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas image pipeline-&-toolchain

Photoshop Generative Fill - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Photoshop Generative Fill defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: do not alter regulated product evidence deceptively.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Photoshop Generative Fill: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-462; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

model-atlas image evaluation-court

Photoshop Generative Fill - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Photoshop Generative Fill uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Photoshop Generative Fill on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-463; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

Photoshop Generative Fill - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Photoshop Generative Fill must be known before deployment: do not alter regulated product evidence deceptively.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Photoshop Generative Fill output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-464; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

Photoshop Generative Fill - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Photoshop Generative Fill stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a Photoshop Generative Fill generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-465; model=Photoshop Generative Fill; confidence=Established; update_status=watch; risk=do not alter regulated product evidence deceptively

Premiere / After Effects AI finishing - Positioning & Ontology

CLAIM / FUNCTION

Premiere / After Effects AI finishing is treated as a bounded production organ, not a universal answer. Type: timeline and motion post-production. Best use: commercial-grade editing, captions, compositing, motion graphics. Epistemic rank: Established.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Premiere / After Effects AI finishing only
if its strength matches the target. State why. Then produce a prompt
with: subject, active difference, layout, palette, typography, camera,
proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-466; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

model-atlas timeline commercial-use-map

Premiere / After Effects AI finishing - Commercial Use Map

CLAIM / FUNCTION

Commercially, Premiere / After Effects AI finishing must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Premiere / After Effects AI finishing, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Premiere / After Effects AI finishing: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-467; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

model-atlas timeline aesthetic-grammar

Premiere / After Effects AI finishing - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Premiere / After Effects AI finishing maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-468; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

Premiere / After Effects AI finishing - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Premiere / After Effects AI finishing is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Premiere / After Effects AI finishing: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-469; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

Premiere / After Effects AI finishing - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Premiere / After Effects AI finishing converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Premiere / After Effects AI finishing: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-470; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

Premiere / After Effects AI finishing - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Premiere / After Effects AI finishing defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: AI is assistant, not final editor; preserve project files.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Premiere / After Effects AI finishing: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-471; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

model-atlas timeline evaluation-court

Premiere / After Effects AI finishing - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Premiere / After Effects AI finishing uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Premiere / After Effects AI finishing on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-472; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

Premiere / After Effects AI finishing - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Premiere / After Effects AI finishing must be known before deployment: AI is assistant, not final editor; preserve project files.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Premiere / After Effects AI finishing output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-473; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

model-atlas timeline rag/graphrag-metadata

Premiere / After Effects AI finishing - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Premiere / After Effects AI finishing stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Premiere / After Effects AI finishing
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-474; model=Premiere / After Effects AI finishing; confidence=Established; update_status=watch; risk=AI is assistant, not final editor; preserve project files

model-atlas video-as-code positioning-&-ontology

Remotion - Positioning & Ontology

CLAIM / FUNCTION

Remotion is treated as a bounded production organ, not a universal answer. Type: video-as-code with React. Best use: template videos, parameterized ads, automated variants, CI renders. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Remotion only if its strength matches the
target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-475; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code commercial-use-map

Remotion - Commercial Use Map

CLAIM / FUNCTION

Commercially, Remotion must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Remotion, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Remotion: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-476; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code aesthetic-grammar

Remotion - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Remotion maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-477; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code prompt-injection-loop

Remotion - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Remotion is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Remotion: Extract Δ. Translate Δ into one visual
paradox. Set typography hierarchy. Generate. Judge with 5 criteria.
Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-478; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code shot/motion-grammar

Remotion - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Remotion converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Remotion: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-479; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code pipeline-&-toolchain

Remotion - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Remotion defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: requires engineering discipline; animation has to be designed.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Remotion: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-480; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code evaluation-court

Remotion - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Remotion uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Remotion on 10 criteria. For each weak point,
write a repair instruction and decide: regenerate, edit, composite, or
reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-481; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

model-atlas video-as-code failure-modes-&-repair

Remotion - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Remotion must be known before deployment: requires engineering discipline; animation has to be designed.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Remotion output: generic style, wrong text,
object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-482; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

Remotion - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Remotion stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Remotion generation: nodes=model,
prompt, style, asset, critique, rights, repair; edges=generates,
constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-483; model=Remotion; confidence=Supported; update_status=watch; risk=requires engineering discipline; animation has to be designed

HyperFrames / HTML-to-video - Positioning & Ontology

CLAIM / FUNCTION

HyperFrames / HTML-to-video is treated as a bounded production organ, not a universal answer. Type: deterministic MP4 rendering from HTML/CSS/JS. Best use: agent-built video, editorial templates, deterministic captions and graphics. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to HyperFrames / HTML-to-video only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-484; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

model-atlas deterministic commercial-use-map

HyperFrames / HTML-to-video - Commercial Use Map

CLAIM / FUNCTION

Commercially, HyperFrames / HTML-to-video must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For HyperFrames / HTML-to-video, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for HyperFrames / HTML-to-video: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-485; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for HyperFrames / HTML-to-video maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-486; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for HyperFrames / HTML-to-video is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For HyperFrames / HTML-to-video: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-487; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

model-atlas deterministic shot/motion-grammar

HyperFrames / HTML-to-video - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for HyperFrames / HTML-to-video converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for HyperFrames / HTML-to-video: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-488; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for HyperFrames / HTML-to-video defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: not a magic film generator; it needs designed composition code.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using HyperFrames / HTML-to-video: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-489; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - Evaluation Court

CLAIM / FUNCTION

Evaluation court for HyperFrames / HTML-to-video uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of HyperFrames / HTML-to-video on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-490; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for HyperFrames / HTML-to-video must be known before deployment: not a magic film generator; it needs designed composition code.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on HyperFrames / HTML-to-video output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-491; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

HyperFrames / HTML-to-video - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for HyperFrames / HTML-to-video stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a HyperFrames / HTML-to-video generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-492; model=HyperFrames / HTML-to-video; confidence=Supported; update_status=watch; risk=not a magic film generator; it needs designed composition code

model-atlas vector positioning-&-ontology

Rive / Lottie motion - Positioning & Ontology

CLAIM / FUNCTION

Rive / Lottie motion is treated as a bounded production organ, not a universal answer. Type: vector animation systems. Best use: micro-animations, app/product explainer motion, reusable brand movement. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Rive / Lottie motion only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-493; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Commercial Use Map

CLAIM / FUNCTION

Commercially, Rive / Lottie motion must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Rive / Lottie motion, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Rive / Lottie motion: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-494; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Rive / Lottie motion maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-495; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Rive / Lottie motion is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Rive / Lottie motion: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-496; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Rive / Lottie motion converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNE

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Shot list for Rive / Lottie motion: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-497; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Rive / Lottie motion defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: complex character/cinematic motion requires different tools.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Rive / Lottie motion: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-498; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Rive / Lottie motion uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Rive / Lottie motion on 10 criteria. For each
weak point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-499; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Rive / Lottie motion must be known before deployment: complex character/cinematic motion requires different tools.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Rive / Lottie motion output: generic style,
wrong text, object drift, rights risk, uncanny motion, overclaim, weak
hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-500; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Rive / Lottie motion - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Rive / Lottie motion stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a Rive / Lottie motion generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-501; model=Rive / Lottie motion; confidence=Established; update_status=watch; risk=complex character/cinematic motion requires different tools

Blender + AI extensions - Positioning & Ontology

CLAIM / FUNCTION

Blender + AI extensions is treated as a bounded production organ, not a universal answer. Type: 3D production and procedural rendering. Best use: product physics, camera realism, 3D assets, cinematic control. Epistemic rank: Established.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to Blender + AI extensions only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-502; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Commercial Use Map

CLAIM / FUNCTION

Commercially, Blender + AI extensions must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Blender + AI extensions, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Blender + AI extensions: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-503; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Blender + AI extensions maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-504; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

model-atlas | 3D | prompt-injection-loop

Blender + AI extensions - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Blender + AI extensions is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Blender + AI extensions: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-505; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Blender + AI extensions converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Blender + AI extensions: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-506; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Blender + AI extensions defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: requires real 3D skill; AI shortcuts do not replace topology/material QA.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using Blender + AI extensions: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-507; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Blender + AI extensions uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Blender + AI extensions on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-508; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Blender + AI extensions must be known before deployment: requires real 3D skill; AI shortcuts do not replace topology/material QA.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Blender + AI extensions output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-509; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

Blender + AI extensions - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Blender + AI extensions stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Blender + AI extensions generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-510; model=Blender + AI extensions; confidence=Established; update_status=watch; risk=requires real 3D skill; AI shortcuts do not replace topology/material QA

model-atlas real-time positioning-&-ontology

Unreal Engine virtual production - Positioning & Ontology

CLAIM / FUNCTION

Unreal Engine virtual production is treated as a bounded production organ, not a universal answer. Type: real-time 3D cinematics. Best use: virtual sets, product worlds, previsualization, cinematic lighting. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Unreal Engine virtual production only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-511; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time commercial-use-map

Unreal Engine virtual production - Commercial Use Map

CLAIM / FUNCTION

Commercially, Unreal Engine virtual production must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Unreal Engine virtual production, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Unreal Engine virtual production: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-512; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time aesthetic-grammar

Unreal Engine virtual production - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Unreal Engine virtual production maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-513; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time prompt-injection-loop

Unreal Engine virtual production - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Unreal Engine virtual production is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Unreal Engine virtual production: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-514; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

Unreal Engine virtual production - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Unreal Engine virtual production converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Unreal Engine virtual production: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-515; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time pipeline-&-toolchain

Unreal Engine virtual production - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Unreal Engine virtual production defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: production pipeline complexity and asset rights.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Unreal Engine virtual production:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-516; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time evaluation-court

Unreal Engine virtual production - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Unreal Engine virtual production uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Unreal Engine virtual production on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-517; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

Unreal Engine virtual production - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Unreal Engine virtual production must be known before deployment: production pipeline complexity and asset rights.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on Unreal Engine virtual production output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-518; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas real-time rag/graphrag-metadata

Unreal Engine virtual production - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Unreal Engine virtual production stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Unreal Engine virtual production
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-519; model=Unreal Engine virtual production; confidence=Established; update_status=watch; risk=production pipeline complexity and asset rights

model-atlas audio positioning-&-ontology

ElevenLabs Speech/SFX/Music - Positioning & Ontology

CLAIM / FUNCTION

ElevenLabs Speech/SFX/Music is treated as a bounded production organ, not a universal answer. Type: audio generation and localization. Best use: voiceover, sound effects, dubbing, music/ambient audio for ads. Epistemic rank: Supported.

OPERATIONAL KERNE

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
SYSTEM: Route the brief to ElevenLabs Speech/SFX/Music only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-520; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio commercial-use-map

ElevenLabs Speech/SFX/Music - Commercial Use Map

CLAIM / FUNCTION

Commercially, ElevenLabs Speech/SFX/Music must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For ElevenLabs Speech/SFX/Music, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for ElevenLabs Speech/SFX/Music: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-521; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio aesthetic-grammar

ElevenLabs Speech/SFX/Music - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for ElevenLabs Speech/SFX/Music maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-522; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio prompt-injection-loop

ElevenLabs Speech/SFX/Music - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for ElevenLabs Speech/SFX/Music is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For ElevenLabs Speech/SFX/Music: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-523; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio shot/motion-grammar

ElevenLabs Speech/SFX/Music - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for ElevenLabs Speech/SFX/Music converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for ElevenLabs Speech/SFX/Music: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-524; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio pipeline-&-toolchain

ElevenLabs Speech/SFX/Music - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for ElevenLabs Speech/SFX/Music defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: voice consent, commercial rights and fraud prevention.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using ElevenLabs Speech/SFX/Music: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-525; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio evaluation-court

ElevenLabs Speech/SFX/Music - Evaluation Court

CLAIM / FUNCTION

Evaluation court for ElevenLabs Speech/SFX/Music uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of ElevenLabs Speech/SFX/Music on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-526; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

ElevenLabs Speech/SFX/Music - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for ElevenLabs Speech/SFX/Music must be known before deployment: voice consent, commercial rights and fraud prevention.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on ElevenLabs Speech/SFX/Music output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-527; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas audio rag/graphrag-metadata

ElevenLabs Speech/SFX/Music - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for ElevenLabs Speech/SFX/Music stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a ElevenLabs Speech/SFX/Music generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-528; model=ElevenLabs Speech/SFX/Music; confidence=Supported; update_status=watch; risk=voice consent, commercial rights and fraud prevention

model-atlas avatar positioning-&-ontology

HeyGen avatars - Positioning & Ontology

CLAIM / FUNCTION

HeyGen avatars is treated as a bounded production organ, not a universal answer. Type: avatar video and talking presenter. Best use: education, sales, multilingual explainers, UGC-like presenter videos. Epistemic rank: Supported.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to HeyGen avatars only if its strength matches
the target. State why. Then produce a prompt with: subject, active
difference, layout, palette, typography, camera, proof constraint,
negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-529; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

HeyGen avatars - Commercial Use Map

CLAIM / FUNCTION

Commercially, HeyGen avatars must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For HeyGen avatars, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for HeyGen avatars: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-530; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

HeyGen avatars - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for HeyGen avatars maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-531; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

model-atlas avatar prompt-injection-loop

HeyGen avatars - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for HeyGen avatars is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For HeyGen avatars: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-532; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

HeyGen avatars - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for HeyGen avatars converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for HeyGen avatars: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-533; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

model-atlas avatar pipeline-&-toolchain

HeyGen avatars - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for HeyGen avatars defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: likeness consent and disclosure are mandatory.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using HeyGen avatars: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-534; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

HeyGen avatars - Evaluation Court

CLAIM / FUNCTION

Evaluation court for HeyGen avatars uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of HeyGen avatars on 10 criteria. For each weak
point, write a repair instruction and decide: regenerate, edit,
composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-535; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

model-atlas avatar failure-modes-&-repair

HeyGen avatars - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for HeyGen avatars must be known before deployment: likeness consent and disclosure are mandatory.

OPERATIONAL KERNE

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Run failure diagnosis on HeyGen avatars output: generic style, wrong
text, object drift, rights risk, uncanny motion, overclaim, weak hook.
Propose the fastest repair path.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-536; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

model-atlas avatar rag/graphrag-metadata

HeyGen avatars - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for HeyGen avatars stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a HeyGen avatars generation: nodes=model, prompt, style, asset, critique, rights, repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-537; model=HeyGen avatars; confidence=Supported; update_status=watch; risk=likeness consent and disclosure are mandatory

model-atlas enterprise positioning-&-ontology

Synthesia avatars - Positioning & Ontology

CLAIM / FUNCTION

Synthesia avatars is treated as a bounded production organ, not a universal answer. Type: enterprise avatar video. Best use: training, corporate explainers, scripted multilingual content. Epistemic rank: Established/verify current.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Synthesia avatars only if its strength
matches the target. State why. Then produce a prompt with: subject,
active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-538; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

Synthesia avatars - Commercial Use Map

CLAIM / FUNCTION

Commercially, Synthesia avatars must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Synthesia avatars, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Synthesia avatars: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-539; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

Synthesia avatars - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Synthesia avatars maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-540; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

Synthesia avatars - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Synthesia avatars is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNE

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
[HYPERPROMPT] For Synthesia avatars: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-541; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

Synthesia avatars - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Synthesia avatars converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Synthesia avatars: 0-2s hook, 2-5s surreal proof, 5-8s
product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned
with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-542; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

Synthesia avatars - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Synthesia avatars defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: avatar fatigue; not suitable for every brand tone.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Synthesia avatars: inputs, model
settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-543; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

model-atlas enterprise evaluation-court

Synthesia avatars - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Synthesia avatars uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Synthesia avatars on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-544; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

model-atlas enterprise failure-modes-&-repair

Synthesia avatars - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Synthesia avatars must be known before deployment: avatar fatigue; not suitable for every brand tone.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Synthesia avatars output: generic style, wrong text, object drift, rights risk, uncanny motion, overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-545; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

model-atlas enterprise rag/graphrag-metadata

Synthesia avatars - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Synthesia avatars stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Synthesia avatars generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-546; model=Synthesia avatars; confidence=Established/verify current; update_status=watch; risk=avatar fatigue; not suitable for every brand tone

model-atlas audio/video positioning-&-ontology

Descript / transcript-first editing - Positioning & Ontology

CLAIM / FUNCTION

Descript / transcript-first editing is treated as a bounded production organ, not a universal answer. Type: audio/video editing workflow. Best use: podcast clips, UGC cleanup, voiceover edit, captions. Epistemic rank: Established/verify current.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Descript / transcript-first editing only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-547; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video commercial-use-map

Descript / transcript-first editing - Commercial Use Map

CLAIM / FUNCTION

Commercially, Descript / transcript-first editing must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For Descript / transcript-first editing, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for Descript / transcript-first editing: 1 big
title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1
color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-548; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video aesthetic-grammar

Descript / transcript-first editing - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Descript / transcript-first editing maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-549; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video prompt-injection-loop

Descript / transcript-first editing - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Descript / transcript-first editing is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Descript / transcript-first editing: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-550; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

Descript / transcript-first editing - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Descript / transcript-first editing converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Descript / transcript-first editing: 0-2s hook, 2-5s surreal proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio: aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-551; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video pipeline-&-toolchain

Descript / transcript-first editing - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Descript / transcript-first editing defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: transcript edits can alter meaning; final listen required.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Descript / transcript-first editing:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-552; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video evaluation-court

Descript / transcript-first editing - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Descript / transcript-first editing uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Descript / transcript-first editing on 10
criteria. For each weak point, write a repair instruction and decide:
regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-553; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

model-atlas audio/video failure-modes-&-repair

Descript / transcript-first editing - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Descript / transcript-first editing must be known before deployment: transcript edits can alter meaning; final listen required.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Descript / transcript-first editing output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-554; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

Descript / transcript-first editing - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Descript / transcript-first editing stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Descript / transcript-first editing
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-555; model=Descript / transcript-first editing; confidence=Established/verify current; update_status=watch; risk=transcript edits can alter meaning; final listen required

DaVinci Resolve / Fusion - Positioning & Ontology

CLAIM / FUNCTION

DaVinci Resolve / Fusion is treated as a bounded production organ, not a universal answer. Type: professional color/edit/compositing. Best use: final color, grade, sound, VFX, quality control. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to DaVinci Resolve / Fusion only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-556; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Commercial Use Map

CLAIM / FUNCTION

Commercially, DaVinci Resolve / Fusion must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For DaVinci Resolve / Fusion, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for DaVinci Resolve / Fusion: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-557; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for DaVinci Resolve / Fusion maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNEL

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-558; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

model-atlas professional prompt-injection-loop

DaVinci Resolve / Fusion - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for DaVinci Resolve / Fusion is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For DaVinci Resolve / Fusion: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-559; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for DaVinci Resolve / Fusion converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for DaVinci Resolve / Fusion: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-560; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for DaVinci Resolve / Fusion defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: needs real post-production knowledge.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using DaVinci Resolve / Fusion: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-561; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Evaluation Court

CLAIM / FUNCTION

Evaluation court for DaVinci Resolve / Fusion uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of DaVinci Resolve / Fusion on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-562; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for DaVinci Resolve / Fusion must be known before deployment: needs real post-production knowledge.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on DaVinci Resolve / Fusion output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-563; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

DaVinci Resolve / Fusion - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for DaVinci Resolve / Fusion stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a DaVinci Resolve / Fusion generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-564; model=DaVinci Resolve / Fusion; confidence=Established; update_status=watch; risk=needs real post-production knowledge

model-atlas programmatic positioning-&-ontology

FFmpeg / MoviePy / OpenCV - Positioning & Ontology

CLAIM / FUNCTION

FFmpeg / MoviePy / OpenCV is treated as a bounded production organ, not a universal answer. Type: programmatic video processing. Best use: batch encoding, frame extraction, crops, versioning, evaluation pipelines. Epistemic rank: Established.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel: :

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to FFmpeg / MoviePy / OpenCV only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-565; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Commercial Use Map

CLAIM / FUNCTION

Commercially, FFmpeg / MoviePy / OpenCV must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For FFmpeg / MoviePy / OpenCV, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for FFmpeg / MoviePy / OpenCV: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-566; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for FFmpeg / MoviePy / OpenCV maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-567; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for FFmpeg / MoviePy / OpenCV is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For FFmpeg / MoviePy / OpenCV: Extract Δ. Translate Δ
into one visual paradox. Set typography hierarchy. Generate. Judge with
5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-568; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for FFmpeg / MoviePy / OpenCV converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for FFmpeg / MoviePy / OpenCV: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-569; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for FFmpeg / MoviePy / OpenCV defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: wrong codecs/settings break delivery specs.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using FFmpeg / MoviePy / OpenCV: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-570; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - Evaluation Court

CLAIM / FUNCTION

Evaluation court for FFmpeg / MoviePy / OpenCV uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNE

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Evaluate the output of FFmpeg / MoviePy / OpenCV on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-571; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

model-atlas programmatic failure-modes-&-repair

FFmpeg / MoviePy / OpenCV - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for FFmpeg / MoviePy / OpenCV must be known before deployment: wrong codecs/settings break delivery specs.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on FFmpeg / MoviePy / OpenCV output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-572; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

FFmpeg / MoviePy / OpenCV - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for FFmpeg / MoviePy / OpenCV stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a FFmpeg / MoviePy / OpenCV generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-573; model=FFmpeg / MoviePy / OpenCV; confidence=Established; update_status=watch; risk=wrong codecs/settings break delivery specs

MLLM Judge + VBench stack - Positioning & Ontology

CLAIM / FUNCTION

MLLM Judge + VBench stack is treated as a bounded production organ, not a universal answer. Type: evaluation and QA layer. Best use: prompt adherence, temporal consistency, compositionality, safety review. Epistemic rank: Supported as evaluation method.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to MLLM Judge + VBench stack only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-574; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Commercial Use Map

CLAIM / FUNCTION

Commercially, MLLM Judge + VBench stack must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For MLLM Judge + VBench stack, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for MLLM Judge + VBench stack: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-575; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for MLLM Judge + VBench stack maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-576; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for MLLM Judge + VBench stack is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For MLLM Judge + VBench stack: Extract Δ. Translate Δ
into one visual paradox. Set typography hierarchy. Generate. Judge with
5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-577; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for MLLM Judge + VBench stack converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for MLLM Judge + VBench stack: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-578; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for MLLM Judge + VBench stack defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: judges are not truth; calibrate with human review.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using MLLM Judge + VBench stack: inputs,
model settings, reference files, post tools, QA gates, export specs,
metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-579; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Evaluation Court

CLAIM / FUNCTION

Evaluation court for MLLM Judge + VBench stack uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of MLLM Judge + VBench stack on 10 criteria. For each weak point, write a repair instruction and decide: regenerate, edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-580; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for MLLM Judge + VBench stack must be known before deployment: judges are not truth; calibrate with human review.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on MLLM Judge + VBench stack output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-581; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

MLLM Judge + VBench stack - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for MLLM Judge + VBench stack stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a MLLM Judge + VBench stack generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-582; model=MLLM Judge + VBench stack; confidence=Supported as evaluation method; update_status=watch; risk=judges are not truth; calibrate with human review

GraphRAG creative memory - Positioning & Ontology

CLAIM / FUNCTION

GraphRAG creative memory is treated as a bounded production organ, not a universal answer. Type: retrieval and style-governance layer. Best use: brand memory, prompt reuse, style constraints, evidence governance. Epistemic rank: Theoretical synthesis.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to GraphRAG creative memory only if its
strength matches the target. State why. Then produce a prompt with:
subject, active difference, layout, palette, typography, camera, proof
constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-583; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

GraphRAG creative memory - Commercial Use Map

CLAIM / FUNCTION

Commercially, GraphRAG creative memory must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNEL

The conversion law: style earns its place only when it lowers friction or raises trust. For GraphRAG creative memory, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create an ad concept for GraphRAG creative memory: 1 big title, 1
strange metaphor, 1 proof object, 1 call-to-action area, 1 color
tension. Avoid generic futuristic grids.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-584; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

model-atlas retrieval aesthetic-grammar

GraphRAG creative memory - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for GraphRAG creative memory maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette luxury black/gold/white; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A luxury black/gold/white conceptual commercial visual: ordinary object
+ impossible knowledge-machine metaphor + huge classy title + clean
flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-585; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

model-atlas retrieval prompt-injection-loop

GraphRAG creative memory - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for GraphRAG creative memory is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For GraphRAG creative memory: Extract Δ. Translate Δ into one visual paradox. Set typography hierarchy. Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-586; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

GraphRAG creative memory - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for GraphRAG creative memory converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for GraphRAG creative memory: 0-2s hook, 2-5s surreal proof,
5-8s product clarity, 8-10s CTA. Camera motion: restrained. Audio:
aligned with motion. Repair drift by re-seeding reference frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-587; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

GraphRAG creative memory - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for GraphRAG creative memory defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNEL

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: stale memories can fossilize mistakes.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Build production pipeline using GraphRAG creative memory: inputs, model settings, reference files, post tools, QA gates, export specs, metadata, rollback plan.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-588; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

GraphRAG creative memory - Evaluation Court

CLAIM / FUNCTION

Evaluation court for GraphRAG creative memory uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: turn one abstract claim into a spatial relation, a texture, and a motion cue. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of GraphRAG creative memory on 10 criteria. For
each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-589; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

GraphRAG creative memory - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for GraphRAG creative memory must be known before deployment: stale memories can fossilize mistakes.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: convert style into a repeatable brand token rather than a one-off mood. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on GraphRAG creative memory output: generic
style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-590; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

GraphRAG creative memory - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for GraphRAG creative memory stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNE

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: compress complexity into a visible, clickable, commercial symbol. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create a GraphRAG record for a GraphRAG creative memory generation:
nodes=model, prompt, style, asset, critique, rights, repair;
edges=generates, constrains, tests, updates.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-591; model=GraphRAG creative memory; confidence=Theoretical synthesis; update_status=watch; risk=stale memories can fossilize mistakes

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

model-atlas prompt positioning-&-ontology

Moussaif HyperPrompt Loop Engine - Positioning & Ontology

CLAIM / FUNCTION

Moussaif HyperPrompt Loop Engine is treated as a bounded production organ, not a universal answer. Type: prompt orchestration OS. Best use: brief-to-shot-to-render-to-judge-to-repair loops. Epistemic rank: Theoretical synthesis.

OPERATIONAL KERNEL

Ontology: this slot is a process-plus-tool relation. It receives a brief, constraints and references; it outputs candidate assets, not final truth. Use it when the required visual difference matches its strongest control surface. Arabic kernel:

.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
SYSTEM: Route the brief to Moussaif HyperPrompt Loop Engine only if its strength matches the target. State why. Then produce a prompt with: subject, active difference, layout, palette, typography, camera, proof constraint, negative constraints.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-592; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Moussaif HyperPrompt Loop Engine - Commercial Use Map

CLAIM / FUNCTION

Commercially, Moussaif HyperPrompt Loop Engine must be assigned to a conversion job: attention hook, proof scene, product desire, educational explainer, or brand world-building.

OPERATIONAL KERNE

The conversion law: style earns its place only when it lowers friction or raises trust. For Moussaif HyperPrompt Loop Engine, the ad must define where the eye lands first, what claim becomes visible, and what cannot be exaggerated.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Create an ad concept for Moussaif HyperPrompt Loop Engine: 1 big title, 1 strange metaphor, 1 proof object, 1 call-to-action area, 1 color tension. Avoid generic futuristic grids.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-593; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Moussaif HyperPrompt Loop Engine - Aesthetic Grammar

CLAIM / FUNCTION

Aesthetic grammar for Moussaif HyperPrompt Loop Engine maps the model/tool to visual strategies: minimal, flat, pop art, patchwork, editorial realism, or neo-tech abstraction.

OPERATIONAL KERNE

Grammar rule: choose palette pop-art cyan/yellow/magenta; combine one ordinary product with one impossible symbolic environment; keep hierarchy clear; make text legible; preserve commercial function.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
A pop-art cyan/yellow/magenta conceptual commercial visual: ordinary
object + impossible knowledge-machine metaphor + huge classy title +
clean flat/popup/patchwork layering + no clutter + premium ad finish.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-594; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

model-atlas prompt prompt-injection-loop

Moussaif HyperPrompt Loop Engine - Prompt Injection Loop

CLAIM / FUNCTION

Prompt injection for Moussaif HyperPrompt Loop Engine is not random wording; it is a loop that injects constraints, tests, and repair rules into the creative request.

OPERATIONAL KERNEL

HyperPrompt loop: Δ-extract -> metaphor -> style tokens -> control tokens -> negative constraints -> QA questions -> repair prompt. The model receives not only desire, but a criticism-ready task.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
[HYPERPROMPT] For Moussaif HyperPrompt Loop Engine: Extract Δ.
Translate Δ into one visual paradox. Set typography hierarchy.
Generate. Judge with 5 criteria. Rewrite prompt to repair the weakest
criterion.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-595; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Moussaif HyperPrompt Loop Engine - Shot/Motion Grammar

CLAIM / FUNCTION

Shot and motion grammar for Moussaif HyperPrompt Loop Engine converts static meaning into time: reveal, collision, transformation, proof, emotional release, and CTA landing.

OPERATIONAL KERNEL

Every video shot must name subject, action, camera, duration, transition, audio cue and failure risk. If a model cannot hold continuity, reduce ambition and split the scene.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Shot list for Moussaif HyperPrompt Loop Engine: 0-2s hook, 2-5s surreal
proof, 5-8s product clarity, 8-10s CTA. Camera motion: restrained.
Audio: aligned with motion. Repair drift by re-seeding reference
frames.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-596; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Moussaif HyperPrompt Loop Engine - Pipeline & Toolchain

CLAIM / FUNCTION

Pipeline for Moussaif HyperPrompt Loop Engine defines how it enters the agency box: brief -> prompt -> generation -> edit -> post -> evaluate -> store -> deploy.

OPERATIONAL KERNE

Toolchain law: generation is only one node. Surround it with design systems, masks, editors, upscalers, encoding, version logs, and legal checks. Risk: can become decorative unless every loop has output and repair criteria.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTIO

```
Build production pipeline using Moussaif HyperPrompt Loop Engine:
inputs, model settings, reference files, post tools, QA gates, export
specs, metadata, rollback plan.
```

DESTRUCTION COUR

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-597; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

model-atlas prompt evaluation-court

Moussaif HyperPrompt Loop Engine - Evaluation Court

CLAIM / FUNCTION

Evaluation court for Moussaif HyperPrompt Loop Engine uses qualitative scoring inspired by VBench/T2I/T2V families without pretending fake precision.

OPERATIONAL KERNEL

Judge: prompt adherence, text legibility, composition, temporal consistency if video, causality, identity stability, brand safety, rights status, and conversion relevance. Human review remains final.

MECHANISM

Mechanism: separate invention, execution, evaluation and rights review into different agents. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Evaluate the output of Moussaif HyperPrompt Loop Engine on 10 criteria.
For each weak point, write a repair instruction and decide: regenerate,
edit, composite, or reject.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-598; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

model-atlas prompt failure-modes-&-repair

Moussaif HyperPrompt Loop Engine - Failure Modes & Repair

CLAIM / FUNCTION

Failure modes for Moussaif HyperPrompt Loop Engine must be known before deployment: can become decorative unless every loop has output and repair criteria.

OPERATIONAL KERNEL

Repair is model-specific: reduce scene complexity, add references, split shots, lock text in post, use masks, version seeds, and use a different model when the failure is structural.

MECHANISM

Mechanism: make the hidden variable visible before selecting a style. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Run failure diagnosis on Moussaif HyperPrompt Loop Engine output:
generic style, wrong text, object drift, rights risk, uncanny motion,
overclaim, weak hook. Propose the fastest repair path.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-599; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.

Moussaif HyperPrompt Loop Engine - RAG/GraphRAG Metadata

CLAIM / FUNCTION

RAG/GraphRAG metadata for Moussaif HyperPrompt Loop Engine stores the model slot as a reusable knowledge chunk for future campaigns.

OPERATIONAL KERNEL

Metadata captures model, task, style tokens, prompt, negative prompt, references, rights, QA score, platform, failure, repair and update status. Memory improves only if bad outputs are logged too.

MECHANISM

Mechanism: route the prompt through audience state, platform format, and proof demand. Boundary: do not promise unsupported features; verify access, license and output quality before production.

HYPERPROMPT INJECTION

```
Create a GraphRAG record for a Moussaif HyperPrompt Loop Engine
generation: nodes=model, prompt, style, asset, critique, rights,
repair; edges=generates, constrains, tests, updates.
```

DESTRUCTION COURT

Destruction court: hallucination check, rights check, text check, style-function check, audience-action check. The page fails if the output cannot be evaluated or repaired. RAG metadata: chunk_id=V3-MDL-600; model=Moussaif HyperPrompt Loop Engine; confidence=Theoretical synthesis; update_status=watch; risk=can become decorative unless every loop has output and repair criteria

Source governance: current public docs and benchmark families are used for orientation only. Re-check pricing, region, license, model version and API surface before client deployment.