Search Beyond What Can Be Taught: Evolving the Knowledge Boundary in Agentic Visual Generation

Haozhe Wang[1] **Weijia Feng**[3] **Jinpeng Yu**[3] **Che Liu**[4]
Ping Nie[2] **Fangzhen Lin**[1] **Jiaming Liu**[3,✉] **Ruihua Huang**[3]
Jimmy Lin[2] **Wenhu Chen**[2] **Cong Wei**[2,✉]

[1]Hong Kong University of Science and Technology [2]University of Waterloo
[3]Qwen Applications [4]Imperial College London

Project Page: https://haozheh3.github.io/SearchGen

Abstract

Visual generators excel at rendering, but they confidently fabricate what they do not know. User requests are unbounded, evolving, and deeply long-tailed: new characters, trending entities, post-cutoff events, etc. This *world-knowledge bottleneck* is structural: generators are trained on fixed corpora, but the visual world is open-ended. We construct **SEARCHGEN-20K** and **SEARCHGEN-BENCH**, 20,839 prompts spanning twelve failure categories and twenty-two domains, paired with a pre-executed multimodal **SEARCHGEN-CORPUS-1M** to facilitate offline, reproducible research. On SEARCHGEN-BENCH, frontier open generators score only 21–28 out of 100, a 40-point collapse invisible to existing benchmarks.

The natural remedy to this knowledge bottleneck is to employ search tools, enabling *agentic visual generation*. But we found that naive search fails: it retrieves indiscriminately, injecting noise into prompts the generator already handles. We trace the root cause to a *generator-specific, evolving knowledge boundary* — the divide between what a generator can internalize through training and what must remain in external context — and show that this boundary, though hard to specify in advance, is discoverable through a teach-then-search co-training framework. Even a minimal recipe of this co-training produces monotonic improvement, laying the foundation for recursive self-improvement of visual generation that meets world-knowledge grounded requests. **We release the full dataset, co-training corpus, and search corpus** as a replayable harness for tool-augmented, world-knowledge-grounded visual generation.

1 Introduction

Ask a frontier image generator for the mascot of the 2025 Osaka Expo: you get a polished, confident fabrication. Ask for a historically accurate Spartan phalanx: you get anachronistic armor rendered in exquisite detail. Modern generators produce complex scenes with precise lighting and coherent structure, saturating on standard benchmarks [27, 5, 13], yet they still fail on a substantial class of real-world user requests (Figure 1). These failures reflect a *world-knowledge* bottleneck rather than a visual-synthesis bottleneck. Visual generators are trained on fixed corpora with inherent knowledge cutoffs, whereas user requests are *unbounded, evolving, and deeply long-tailed*: new characters, regional cultural symbols, niche typography, historical artifacts, and recent events. The world knowledge bottleneck motivates external knowledge access as a natural complement to visual generation, analogous to illustrators consulting references before depicting unfamiliar concepts.

✉: Corresponding Authors.

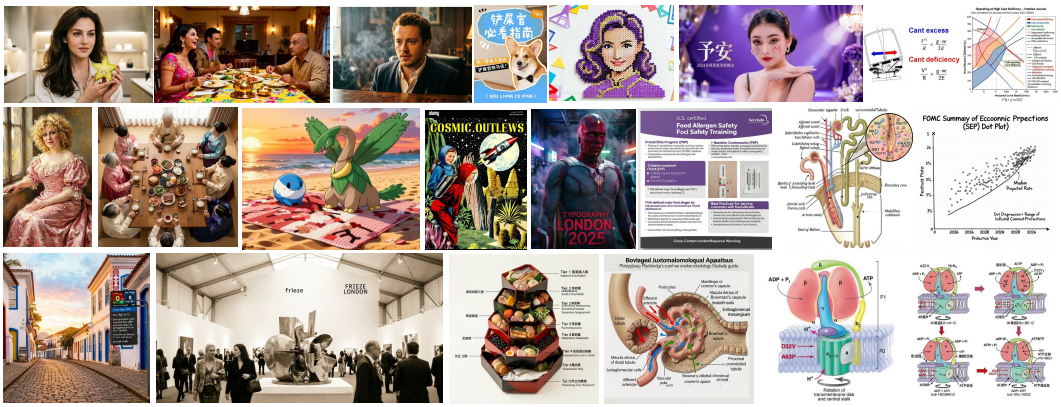

Figure 1: **Representative search-augmented generations from SEARCHGEN-20K, spanning all twelve failure categories identified from 20,840 production prompts.** SEARCHGEN-20K captures the production-scale diverse user requests that demand the unbounded, evolving, and deeply long-tailed world knowledge. The world knowledge ranges from entities that benefit from visual shortcut (left), to complex system or scientific procedures that require textual knowledge. The diversity facilitates research in the structural question this paper addresses: which knowledge gaps can a generator internalize, and which require search at inference time?

To systematically study this problem, we analyze over 20,000 user prompts from production-level AIGC platforms [46] and identify twelve major failure modes. Based on these requests and failure modes, we construct SEARCHGEN-20K, a collection of world-knowledge-grounded prompts spanning twenty-two domains and diverse global cultures, each annotated with fine-grained visual checklists for fully automated assessment (§2). On SEARCHGEN-BENCH, frontier open generators achieve only 21–28 out of 100, at most half of the commercial API ceiling. These results reveal a substantial evaluation gap: existing benchmarks largely fail to capture the live and dynamic world-knowledge failures in visual generation.

Can search guide generators as references guide illustrators?

This renders an agentic visual generation problem: a generator is equipped with an agentic reasoner that decides when and how to utilize search tools for knowledge acquisition. Through extensive experiments, our results show that naive search fails: agentic reasoner that triggers a search blindly for every prompt degrades the quality of a substantial fraction of prompts. Why does naive search fail? We trace the root cause to two distinct issues that must be solved jointly.

The first is when to search. The world knowledge required for visual generation splits along a structural axis into what generators can internalize and what must remain in context. Some knowledge is stable and low-dimensional: a character's canonical appearance, a flag's fixed geometry. Once encountered through augmented supervision, such knowledge migrates into generator parameters and search becomes superfluous. Other knowledge is inherently contextual: it evolves faster than retraining cycles, sits too deep in the long tail for reliable learning [36], or requires per-instance compositional reasoning that resists parameterization. The extreme long tail of visual entities in real requests (Figure 4) confirms that this contextual stratum is large. This creates a knowledge boundary between the two regimes, and this boundary is generator-specific and evolving: it shifts as the generator improves, so a search policy that helps a weaker generator can harm a stronger one.

The second is what search returns. Even when search is correctly triggered, every result injects noise into the generator's input space: irrelevant visual detail, stylistic contamination, spurious structure. The open-weight generators we tested treat all conditioning signals as authoritative, with no capacity to distinguish useful reference content from noise.

We address both issues through two interlocking components. First, we introduce a *noise-resistant* agentic reasoner that builds what to render for generator: the reasoner executes a three-stage gate–filter–integrate pipeline that controls when to search, what retrieved content to retain, and how external information is integrated into generation (§3.1). This pipeline builds noise resistance into the reasoner so that noise is suppressed before reaching the generator. Second, we propose to co-train the reasoner with the visual generator: online iterative DPO teaches the generator to internalize world-knowledge that it can absorb and build noise-resistance to imperfect search-augmented inputs, while

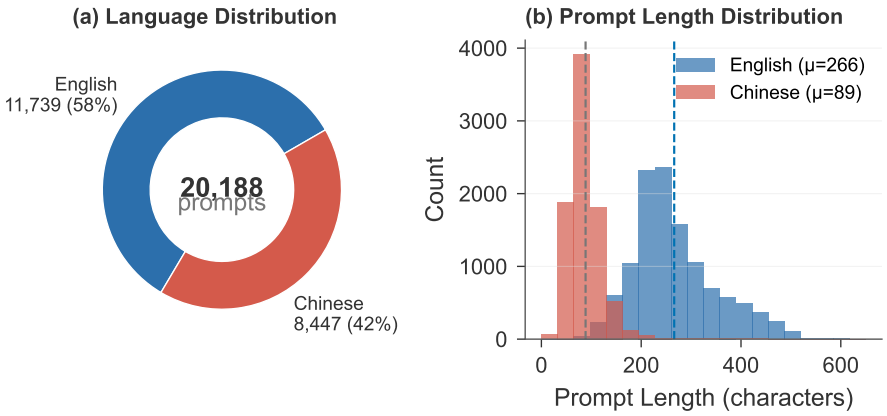

Figure 2: **Bilingual composition and prompt length distribution of SEARCHGEN-20K.** Left: English (58%) and Chinese (42%) proportions. Right: bimodal prompt length distribution – Chinese prompts are concise (mean 89 characters) while English prompts are more elaborate (mean 266 characters), reflecting authentic cross-lingual user behavior rather than translated templates.

rejection-sampling finetuning calibrates the search reasoner to search only for what the generator cannot be taught (§3.3).

On SEARCHGEN-BENCH, this single-iteration recipe produces monotonic gains from NO SEARCH through BLIND SEARCH to GENERATOR-ADAPTIVE SEARCH, exceeding frontier generators paired with frontier VLMs. The recipe is deliberately minimal, yet we envision this co-training framework lays the foundation for a recursive self-improvement flywheel in visual generation: the generator progressively expanding rendering knowledge while the reasoner progressively adjust its search scope, catering to production-scale text-to-image requests that demand dynamically evolving world knowledge.

Contributions.

1. We introduce SEARCHGEN-20K and SEARCHGEN-BENCH, a large-scale dataset of 20,839 world-knowledge-grounded text-to-image prompts spanning twelve failure categories and twenty-two domains, together with a pre-executed multimodal SEARCHGEN-CORPUS-1M for reproducible and democratized research.
2. We show that search-intensive visual generation exposes a substantial evaluation gap: frontier open generators collapse by up to 40 points on SEARCHGEN-BENCH, a gap that shifts as generators improve and that current benchmarks render entirely invisible.
3. We show that naive search introduces *search noise*, causing concept corruption and copy effects, and identify the underlying challenge as a searcher–generator coordination problem governed by a generator-specific knowledge boundary.
4. We propose a co-training framework with a noise-resistant agentic search protocol. We validate that the minimalist recipe, an 8B agentic reasoner with search tools, jointly calibrated with the generator outperforms both pure generation and undiscriminating search by better respecting the knowledge boundary.

2 The World-Knowledge Bottleneck

Faithful generation is limited less by rendering than by *knowledge*. A visual generator is trained on a fixed corpus, but user requests draw on a world that is open-ended, evolving, and long-tailed; the mismatch between the two is the *world-knowledge bottleneck*. The bottleneck is compounded by a structural blind spot: generators are trained to always produce an image, so they have no mechanism for recognizing what they do not know, no way to say "I lack the knowledge to render this faithfully," and no protocol for acquiring it. This section characterizes the bottleneck end to end: we map the landscape of world-knowledge-grounded generation from real user behavior, construct a large-scale dataset and benchmark, quantify how severely it degrades frontier generators, and show that naive search introduces new failures rather than resolving it.

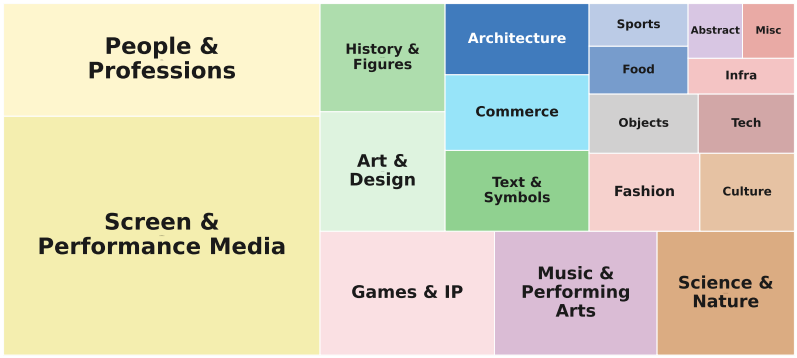

Figure 3: **SEARCHGEN-20K spans diverse, long-tailed domains.** Treemap of benchmark mass across domain categories (area reflects relative prompt counts). The cross-category severity structure between failure modes and domains is deferred to Appendix A.1 (Figure 9), where uniform severity across all cells rules out the hypothesis that failures concentrate in a few niche categories.

Table 1: **Taxonomy of search-intensive visual generation.** Twelve categories organized by failure source, each with a representative example prompt. The *Modality* column indicates the dominant knowledge type required.

Category	Modality	Example Prompt
Temporal – Recent	Both	"The mascot for the 2025 Osaka Expo in its official pose and colors"
Temporal – Current	Both	"Current FIFA World Cup group stage rankings as a scoreboard graphic"
Entity & IP	Visual	"Jingliu from Honkai: Star Rail wielding her ice sword"
Concept & Symbol	Both	"The national flag of Bhutan with the correct Druk dragon design"
Factual & Historical	Both	"Spartan phalanx at Thermopylae with historically accurate bronze armor"
Cultural Specificity	Both	"A traditional Oaxacan alebrije dragon with authentic chromatic patterns"
Visual / UI / UX	Visual	"iOS 17 Weather app screenshot showing a thunderstorm animation"
Data Visualization	Textual	"China dynastic timeline with accurate dates and founding emperors"
Text / Typography	Textual	"Art Nouveau concert poster with period-authentic display lettering"
Complex Composite	Both	"Aztec-style infographic of DNA replication with labeled process stages"
Vague / Abstract	Textual	"The feeling of nostalgia on a rainy afternoon in a small Japanese town"
Implicit Reasoning	Both	"Cozy mountain cabin interior in the style of a Miyazaki film"

2.1 What Users Actually Ask For

To ground the benchmark in real-user demand, we analyzed 20,840 prompts from a production text-to-image platform. The landscape is far more diverse, long-tailed, and rapidly evolving than existing benchmarks suggest. We discover twelve recurring failure categories distilled from real-world user requests (Table 1); Figure 1 shows example generations representative of these failure modes.

As shown in Table 1, the twelve categories split along a structural axis that directly determines search design. Some require *visual* references that resist linguistic specification – a game character's costume silhouette, a national flag's precise color fields – that no textual description can fully specify but that a generator can readily condition on once retrieved. For example, generating *Jingliu from Honkai: Star Rail (a game character)* demands a precise visual identity (costume, weapon, color palette).

Other requests require *textual* knowledge for which no visual shortcut exists or images alone cannot convey. For instance, *a China dynastic timeline with dates and founders* relies entirely on structured factual content (exact names, years, and causal orderings), and *an Aztec-style infographic of DNA replication* requires precise biochemical labels and process orderings for which visual shortcuts rarely exists but encyclopedic text can supply.

Many categories sit in between, requiring both modalities simultaneously: *a traditional Oaxacan alebrije dragon* demands visual references for culturally specific chromatic conventions *and* textual knowledge of what distinguishes an alebrije from a generic dragon; *the mascot for the 2025 Osaka Expo* requires temporally fresh visual references of the character *and* textual context that postdates model training.

Table 2: **We release SEARCHGEN-20K, SEARCHGEN-CORPUS-1M and SEARCHGEN-BENCH which compose replayable environment for agentic visual generation.** These assets provide agentic reasoning and generation traces collected across multiple reasoners and generators for the 20,188 unique training prompts. **SEARCHGEN-CORPUS-1M** provide cached search sessions and downloads which emulate live web search in an offline manner, relieving burden to costly search APIs.

Asset	Scale	Enables
Training Trajectories *(20,188 prompts)*		
Reasoning traces across various frontier reasoners	90,452	search-policy / reward learning
Image Generations using different reasoning trace x generators	281,925	preference & distillation data
Search corpus *(archived, indexed)*		
Search sessions	145,642	reproducible retrieval, no live API
image / web	108,800 / 36,842	
SERP hits (unique URLs)	559,973	retrieval / grounding studies
Cached downloads	370,733	fully offline replay
image / web	259,178 / 111,555	

This dual-modality structure, visual identity on one axis and factual precision on the other, directly motivates the modality-aware search protocol we introduce in §3: neither visual nor textual search alone suffices, and the choice of modality must be made per knowledge gap. In this work, we use Google Search API with both image search and web-text search. Image search and web search are thus two distinct tools the agentic reasoner utilizes between per knowledge gap (§3.1).

2.2 Dataset Construction

From the failure taxonomy (Table. 1), we construct SEARCHGEN-20K spanning all twelve categories and twenty-two domains. The dataset is intentionally bilingual (Figure 2): English prompts (58%) average 266 characters and tend toward descriptive elaboration, while Chinese prompts (42%) average 89 characters and favor dense, implicit specification, capturing authentic cross-lingual prompt behavior rather than translated templates. The full training dataset comprises 20,188 rows (20,187 unique prompts) with an average of 5.2 multimodal knowledge gaps per prompt (90.5% carry three or more), annotated with 34,694 visual reference slots, 16,345 textual knowledge slots. We partition into train (20,188 rows with 128 held out for validation) and test (751) sets.

To construct this dataset, we first extract a seed database of 31,537 entities from production-scale user requests, each annotated with canonical names, estimated training-set frequency, ground-truth visual references (where available), and distinguishing attributes; the distribution is intentionally long-tailed, mirroring real user demand (Figure 4a). With 93.1% of entities appearing in only a single prompt, parametric learning from any feasible training dataset cannot cover this demand; search is structurally unavoidable for the tail. From this seed database, we synthesize the SEARCHGEN-20K dataset via human-brainstormed template-based generation and LLM-assisted rewriting.

During prompt synthesis, we deliberately employ an answer-first strategy. We ask a frontier model to select self-contained world-knowledge gaps based on the seed context. Therefore, each prompt carries *knowledge gap references* by construction, structured slots specifying what visual or textual knowledge the generator is likely to lack, each with a severity label (critical, important, moderate or minimal) and a justification field. These annotations ease the burden of automated evaluation.

Released assets: a replayable harness. Reproducing search-augmented generation normally demands paid API access to both a search engine and a fleet of generators, and results drift as those services change. We remove that barrier. Alongside SEARCHGEN-20K/SEARCHGEN-BENCH, we release the full training dataset: 90,452 reasoning traces collected from different frontier reasoners, and 281,925 image generations spanning multiple reasoner configurations and frontier generators (Table 2). We further release a SEARCHGEN-CORPUS-1M with 145,642 archived image and web search sessions, 559,973 unique search URLs, and 370,733 cached downloads. Because every search is pre-executed and frozen, the pipeline can be replayed without live search API keys or search-result drift, turning an expensive workflow into an offline substrate for preference learning, reward modeling, search-policy design, and retrieval research. Together, the gate–filter–integrate protocol, the co-training loop, and this frozen corpus constitute a reusable *harness* for agentic, tool-augmented visual generation.

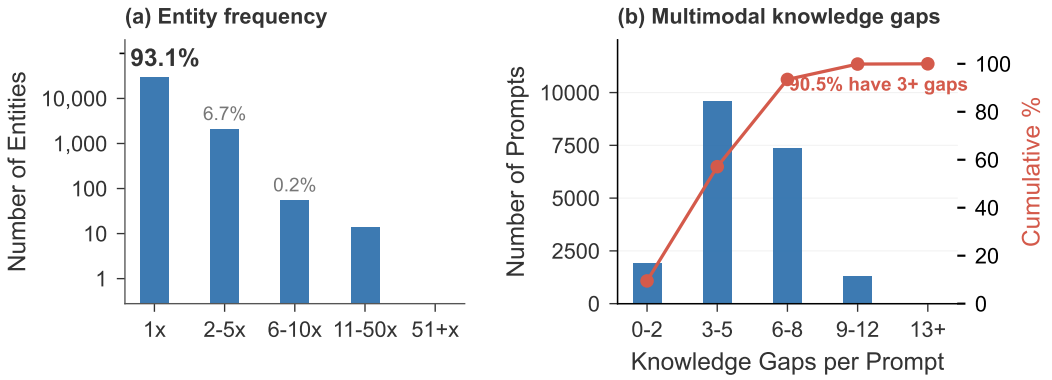

Figure 4: **Dataset composition: entity long-tail and multimodal knowledge gaps. (a)** Entity frequency distribution: 93.1% of the 31,537 unique visual entities appear in only a single prompt, confirming the extreme long-tailed nature of real-world image generation requests. **(b)** Multimodal knowledge gaps: prompts carry a mean of 5.2 simultaneous knowledge gaps (reference slots + text knowledge slots + failure modes); 90.5% carry three or more, confirming the multi-constrained nature of the dataset.

2.3 Evaluation Protocols

Before we deliver our major empirical findings, we first briefly clarify our evaluation protocols for world-knowledge-grounded image generation. We aim to distinguish between world-knowledge failures and rendering failures with fully automated evaluation. Therefore, we employ two complementary set of evaluation score components. As Table 3 details, four components are *knowledge-sensitive*: *Checklist verification* tests 3–5 binary questions about key visual elements. *Rubric-adaptive scoring* applies 2–4 category-tailored weighted dimensions. *Prompt faithfulness* measures overall adherence to the textual request, and *visual reference fidelity* measures faithfulness to provided reference images (when applicable). These evaluation criteria tailor to prompt-specific knowledge demands.

The remaining five components are *knowledge-invariant* dimensions that assess *rendering quality* (bottom half of Table 3): image quality, text rendering, AI naturalness, composition & aesthetics, and physical plausibility. This decomposition comprehensively diagnoses the knowledge bottleneck and rendering gap. We provide an example below for concrete illustration.

Example: Prompt-Specific Evaluation Context.

Prompt: *"Paint Yang Chaoyue (a celebrity) on a 1970s Chinese rural field ridge; long-handled hoe; earth tones; 1970s film grain."*

Checklist (binary verification questions, each targeting one knowledge gap):

- Is the person recognizable as Yang Chaoyue (facial features and overall bearing)?
- Do the clothing and setting match a 1970s Chinese rural commune worker rather than modern workwear?
- Does the farm tool match the form of a traditional long-handled hoe?
- Does the image show the requested earth-tone palette and 1970s film-grain texture?

Rubric (category-specific weighted dimensions; the justification states what each dimension scores):

- `celebrity_likeness` (0.4) — how well the model preserves Yang Chaoyue's face when placed in a non-standard context.
- `historical_authenticity` (0.4) — accuracy of the 1970s attire and farming environment.
- `atmospheric_tone` (0.2) — consistency of the film grain and earth-tone color palette.

Every prompt additionally receives generic dimensions (prompt faithfulness, image quality, AI naturalness, composition).

Knowledge Gap References: Textual knowledge of the background of 1970s Chinese history. Two reference images (period labor clothing; celebrity likeness) supplied alongside the generated image.

End-to-end trace and per-stage reasoner I/O: Appendix D.1.

Table 3: **Evaluation components of the SEARCHGEN-BENCH judge.** Each component is scored 0–100; the overall score is the mean of all applicable components. Knowledge-sensitive components (top) vary per prompt and measure knowledge presence; rendering-quality components (bottom) are fixed across all prompts and measure generation competence independently of knowledge.

Component	What it evaluates
Knowledge-sensitive (prompt-adaptive)	
Checklist Verification	3–10 per-prompt checks on key visual elements, each scored on visual evidence
Rubric Scoring	3–5 weighted scoring scheme that grades quality across a few prompt-adaptive dimensions
Prompt Faithfulness	Presence and accuracy of all requested subjects, attributes, actions, scene, and style
Visual Reference Fidelity	Identity, attribute, and style fidelity to the prompt-requested entities (if applicable)
Textual Knowledge Fidelity	Correctness of factual/textual knowledge gaps (facts, systems, concepts, workflows, etc)
Rendering quality (knowledge-invariant)	
Image Quality	Clarity and sharpness; freedom from artifacts; style, lighting, and perspective coherence
Text Rendering	Accuracy, readability, spelling, and placement of in-image text; scored only when text is required
AI Naturalness	Texture realism vs. AI smoothness and uncanny uniformity; grounded vs. generic look
Composition & Aesthetics	Framing and balance; depth and spatial arrangement; color harmony; overall appeal
Physical Plausibility	Anatomy, object physics, spatial consistency, and material/lighting coherence

2.4 What Makes World-Knowledge-Grounded Image Generation Hard?

> **Finding 1: Generators that score comparably on standard prompts diverge by nearly 40 points when search-intensive world knowledge is required.** On prompts requiring only parametric knowledge, open and commercial generators score at comparable levels (67–75 out of 100); on search-intensive prompts, the same open generators collapse to 22–28 while commercial systems with integrated search barely drop. This 40-point divergence is invisible to existing benchmarks that test only rendering within known concepts.

Figure 5 visualizes the gap directly. On the NoSearch stratum, where only parametric knowledge is required, every generator, open and commercial alike, scores in a comparable band (63–75 out of 100). Generators cluster in a comparable band when knowledge is not the bottleneck. On the Search-Intensive stratum, the landscape fractures: every open-weight generator collapses to 22–28, while commercial systems with integrated search barely drop.

To isolate the world-knowledge bottleneck from rendering ability, we curate two complementary test sets: *NoSearch* (100 prompts selected where generators perform well with parametric knowledge alone) and *Search-Intensive* (651 prompts where external world knowledge is required). Table 4 reports six key evaluation dimensions on the Search-Intensive slice, grouped into knowledge-sensitive components (which measure knowledge presence) and knowledge-invariant components (which measure rendering competence). The full nine-component breakdown including both strata appears in the appendix.

The non-uniformity across commercial systems is itself diagnostic. GPT-Image-2 drops only 0.1 points, because live search operates behind that API. Nano Banana Pro drops only 9.7 points ($75.0 \rightarrow 65.3$), while Qwen-Image-1 drops nearly 40 points. The magnitude of the drop correlates inversely with the degree of integrated search capability: systems with reasoned search maintain performance, e.g., GPT-Image-2, Nano Banana Pro; systems without it collapse, e.g., open-source generators.

Table 4 pinpoints where the gap originates. Knowledge-sensitive components (Checklist, Rubric, Visual Reference Fidelity), which directly measure knowledge presence, collapse across all open generators (e.g., Flux.2-Klein-9B: Checklist 24.2, Visual Reference 16.9). Knowledge-invariant components (Image Quality, Physical Plausibility) remain comparatively high (36.8, 48.6 for the same model), confirming that the collapse reflects knowledge absence, not rendering failure. GPT-Image-2 holds strong across both groups (Checklist 71.2, Visual Reference 66.0), consistent with integrated knowledge access. The benchmark spans diverse, long-tailed domains (Figure 3), and the per-cell severity heatmap (Figure 9, Appendix A.1) confirms the severity is pervasive across all twelve failure categories and twenty-two domains, ruling out the hypothesis that the bottleneck is concentrated in a few niche categories.

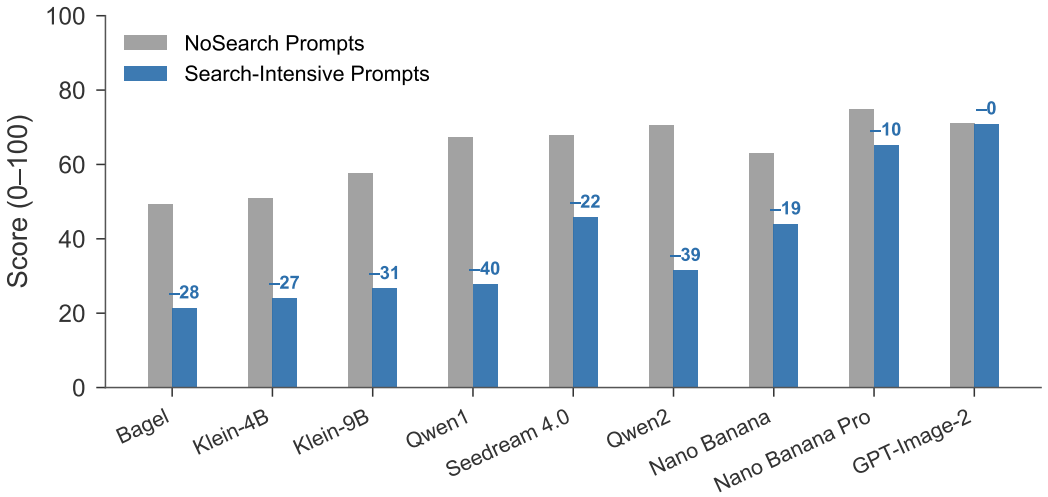

Figure 5: **The world-knowledge bottleneck: per-stratum collapse across nine generators.** Grouped bars show the nine-component mean on NoSearch vs. Search-Intensive strata. Every generator, open-weight and commercial alike, collapses on the search-intensive subset, confirming the bottleneck is universal. Drop magnitudes range from −0.1 (GPT-Image-2) to −39.1 (Qwen-Image-2).

Table 4: **The knowledge bottleneck on SEARCHGEN-BENCH: Search-Intensive subset (651 prompts).** Scores on a 0–100 scale; higher is better. *Knowledge-sensitive* components (Checklist, Rubric, Visual Reference) vary per prompt and directly measure knowledge presence; *knowledge-invariant* components (Text Rendering, Physical Plausibility, Image Quality) measure rendering competence. Knowledge-sensitive components collapse while rendering components hold steady, confirming the failure is knowledge absence, not rendering inability.

	Knowledge-Sensitive			*Knowledge-Invariant*		
Generator	**Checklist**	**Rubric**	**Visual Reference**	**Text**	**Physical**	**Img Quality**
Open-weight generators						
Bagel [12]	18.2	17.6	13.5	2.5	36.8	30.5
Flux.2-Klein-4B [4]	19.8	18.4	11.9	4.2	46.2	37.2
Flux.2-Klein-9B [4]	24.2	23.1	16.9	7.2	48.6	36.8
Qwen-Image [46]	24.8	24.3	17.7	8.7	44.6	40.1
Commercial systems						
Qwen-Image-2 [46]	28.5	27.1	21.0	12.7	48.2	42.2
SeedDream-4.0 [8]	44.2	43.6	35.1	35.9	64.0	57.0
Nano Banana [15]	41.0	40.4	33.2	28.0	65.5	57.1
Nano Banana Pro [15]	64.4	63.1	58.3	65.0	78.5	71.4
GPT-Image-2 [31]	71.2	70.1	66.0	75.9	77.3	75.1

2.5 Naive Search Is a Second Problem

The bottleneck has a natural remedy: search. A human illustrator who receives an unfamiliar request looks up references before drawing. The analogous pipeline for a generator has three steps: (1) a reasoner identifies knowledge gaps in the prompt, (2) search queries fill those gaps, returning images, text, or both, and (3) the retrieved material is integrated into the generator's input so it can render faithfully (Figure 7; §3 formalizes each step).

We stress-test this pipeline with two search policies. *BlindSearch*: the reasoner searches for every nominated gap regardless of whether search actually helps. *ReasonedSearch*: a frontier VLM (Gemini-3-Flash) selectively searches for quality-relevant gaps, then picks the better of the augmented and unaugmented outputs per prompt.

Need Search?	Generator	Baseline	Reasoned	Blind
NoSearch	Qwen2 [46]	70.7	76.5	60.4
	Qwen1 [46]	67.4	75.0	59.5
	Flux.2-Klein-9B [4]	57.8	66.4	52.3
VisualSearch	Qwen2 [46]	37.2	49.1	45.3
	Qwen1 [46]	32.8	44.4	39.7
	Flux.2-Klein-9B [4]	29.6	39.4	36.0
TextualSearch	Qwen2 [46]	22.9	34.1	32.1
	Qwen1 [46]	20.5	28.5	25.3
	Flux.2-Klein-9B [4]	18.9	26.6	24.0

Table 5: **Search is a double-edged sword.** Strata: NoSearch (100), VisualSearch (387), TextualSearch (264), totaling 751 prompts.

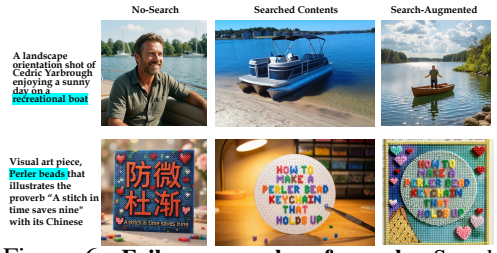

Figure 6: **Failure examples of search.** Search queries executed are colored. Searched contents introduce noise and may degrade generation quality.

To diagnose when and where each policy helps or harms, we further partition the *Search-Intensive* set into two subsets: *VisualSearch* prompt set that benefits substantially from visual references, e.g., celebrities, or *TextualSearch* that benefits from textual references, e.g., scientific infographics.

> **Finding 2: Naive search actively degrades prompts the generator already handles.** BlindSearch drops every generator on the NoSearch stratum (prompts that don't need external world knowledge); even selective search must learn *when* to abstain (Table 5).

If search were uniformly beneficial, the problem would reduce to engineering better queries. Table 5 shows otherwise. On the *NoSearch* stratum, *BlindSearch* strictly degrades every generator: Qwen-Image-2 drops from 70.7 to 60.4, a 14.6% relative loss on prompts it already handles well. *ReasonedSearch* improves Qwen-Image-2 on *NoSearch* (70.7 $\rightarrow$ 76.5) while strongly lifting *TextualSearch* from 22.9 to 34.1. Selective search buys knowledge-intensive gains while preserving parametric knowledge on prompts that do not benefit from retrieval.

Figure 6 illustrates two structurally distinct failure modes. *Concept corruption* (top): search fires on a prompt the generator already handles; the returned reference overrides accurate internal knowledge, a gating failure. *Copy effect* (bottom): a reference carries too much raw information and becomes a copying template rather than a knowledge supplement, a filtering failure. These two modes directly motivate the gate–filter–integrate protocol in §3.1. The generator's inability to filter search noise mirrors text-domain evidence that parametric memory and external lookup trade off by query type [30], but the visual setting adds modality-specific failure modes, including spatial distortion, identity blending, and stylistic leakage, that demand different triage and integration strategies.

3 Co-Training Agentic Visual Generation

The findings above motivate a search-augmented generation paradigm: search only when missing knowledge lies outside the generator's boundary, and control how retrieved content enters generation. This requires deciding *when* to search, *which modality* to use, and *how* to integrate evidence without introducing search noise.

We realize this philosophy through two components (Figure 7): a reasoner controls what to search and how search-augmented inputs (visual references and searched web knowledge) reach the generator with noise resistance (§3.1), and a co-training framework that first teaches the generator to expand knowledge, then recalibrates the reasoner to the strengthened generator's knowledge boundary (§3.3).

3.1 Noise-Resistant Agentic Reasoner

Naive search degrades generation (§2.5) because noise enters at three structurally distinct points: the decision to search, the choice of reference, and the integration of external content. A single filter cannot address all three; each demands its own targeted defense. Therefore, we propose a three-stage agentic search protocol (bottom half of Figure 7) that suppresses noise at each point in sequence (see Appendix D.1 for a worked example). These stages form the agent's reason-and-act loop: *gate* decides when to act and which tool to call, *filter* selects the tool's returns, and *integrate* assimilates the observation into generation.

Stage 1: Gate. In this stage, given a user prompt p, the reasoner identifies knowledge gaps, classifies each by type and severity, and proposes search queries with modality labels (`image` | `web`). Only gaps rated `critical` or `important` trigger search; the rest are filtered, producing at most 3 search queries or SKIP if no actionable gaps survive.

Stage 2: Filter. After multimodal search is executed and search results return, this stage aims to select references that sufficiently fill the specific gap identified while minimizing extraneous content. We seek a reference that "fills the gap" supplies only the missing visual knowledge while leaving the generator free to compose the rest. Empirically, we find that this stage helps reduce the copy effects (§2.5) as visual shortcuts are filtered from search results.

Stage 3: Integrate. Even a well-chosen visual reference can carry far more information than knowledge gap requires, and thus be exploited by the generator as visual shortcuts: raw pixel conditioning leaks style, background, and layout that were never requested. The integration stage therefore *routes visual references through natural language* (Figure 7): the reasoner fuses the original prompt, its gap analysis, retrieved visual and textual knowledge into enriched text specification. In addition to the raw photograph, the generator receives grounded citations naming exactly what to borrow, e.g., "following Image I, render the character in a teal-and-gold robe". This preserves the missing knowledge for faithful generation while discarding the pixel-level noise that would otherwise dominate the output.

3.2 Knowledge Boundary

To address world-knowledge-grounded requests, our key insight is to introduce a reasoner that decides when and what to search, preparing augmented inputs to the generator. Yet this design rests on an implicit assumption: that there exists a well-defined partition between what the generator already knows and what it must search. The observations from §2 confirm this partition is real but also reveal that it is *generator-specific and shifts under training*; search helps some prompts and harms others, and the balance changes with generator capability. This identifies a quantity we define as *knowledge boundary*.

Definition 1 (Knowledge Boundary). Let $\mathcal{K}$ denote the space of world-knowledge units (entities, cultural symbols, typographic identities, procedural specifications, etc.) required by prompts in a distribution $\mathcal{P}$. For a generator G_θ with parameters θ, a prompt $p \in \mathcal{P}$, and conditioning context c (search-returned references and text), let $Q(G_\theta, p, c) \in [0, 1]$ be a bounded quality function. For a fixed tolerance $\epsilon > 0$, define the *internalizable* and *contextual* knowledge sets:

$$\mathcal{K}_{\text{int}}(\theta) = \left\{ k \in \mathcal{K} : \mathbb{E}_{p|k}\left[Q(G_\theta, p, \textsc{Search}(k)) - Q(G_\theta, p, \varnothing)\right] < \epsilon \right\}, \quad (1)$$

$$\mathcal{K}_{\text{ctx}}(\theta) = \mathcal{K} \setminus \mathcal{K}_{\text{int}}(\theta). \quad (2)$$

where the expectation is over all prompts in $\mathcal{P}$ that require knowledge unit k, and $\textsc{Search}(k)$ denotes the conditioning context from executing the search protocol for knowledge unit k. The pair $(\mathcal{K}_{\text{int}}(\theta), \mathcal{K}_{\text{ctx}}(\theta))$ forms a generator-specific partition of $\mathcal{K}$; we refer to this partition as the *knowledge boundary* $\mathcal{B}(\theta)$. A prompt p is *search-intensive* w.r.t. G_θ if it requires knowledge $k \in \mathcal{K}_{\text{ctx}}(\theta)$. The boundary is generator-specific: it depends on θ and shifts under training, i.e., $\mathcal{K}_{\text{int}}(\theta) \subseteq \mathcal{K}_{\text{int}}(\theta')$ when θ' results from DPO on search-augmented demonstrations.

Definition 1 formalizes our fundamental insight.

Insight. Some knowledge is internalizable and search should not fire for it; other knowledge is contextual and search is structurally necessary.

Crucially, the boundary need not be knowable a priori: we find that it is *discoverable* through co-training. Figure 8b makes this concrete: we measure the per-prompt quality gap with and without search and show that its distribution shifts under co-training.

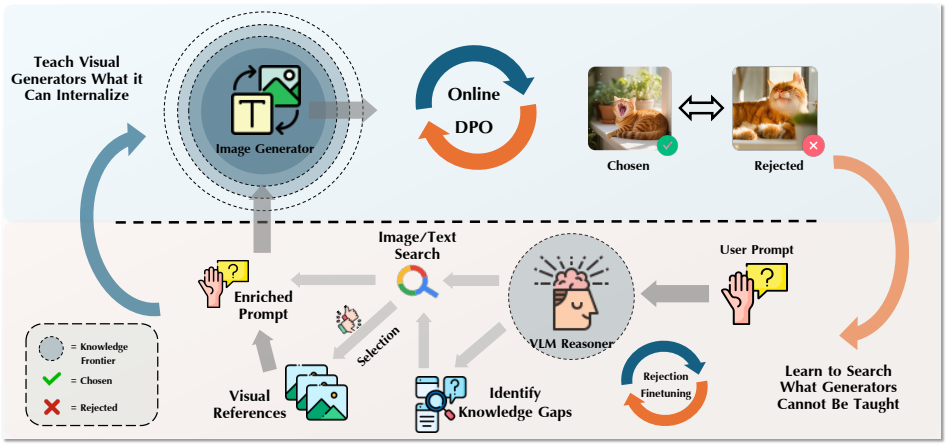

Figure 7: **Co-training framework: teach the generator what it can internalize, then calibrate the reasoner to search what it cannot.** Given a user prompt, the VLM reasoner identifies knowledge gaps, executes modality-aware search (image or text), filters and integrates results into an enriched final prompt, and routes visual references to the generator. Co-training proceeds in two phases. Phase 1 (top): online DPO samples search-augmented generations and ranks them by generation quality, constructing preference pairs to push its knowledge boundary outward. Phase 2 (bottom): rejection-sampling finetuning recalibrates the reasoner to the strengthened generator, reinforcing trajectories where reasoned search improves output and discarding what causes degradation.

3.3 Co-Training: Teach, Then Search

How does co-training discover the boundary in practice? The boundary shifts only when the generator's parameters change, so discovering it requires training the generator itself, not merely adjusting the search policy. Co-training (top half of Figure 7) acts on both sides in sequence: it first strengthens the generator so fewer prompts require search, then recalibrates the reasoner to the strengthened generator's boundary.

Phase 0: Supervised warm-start. Co-training requires a reasoner that follow the prescribed three-stage protocol to reason and search. So we first warm-up the reasoner by supervised finetuning Qwen3-VL-8B [**?**] on ∼10,000 expert-annotated Task A/B/C trajectories, which correspond one-to-one to the gate/filter/integrate stages (representative per-stage I/O in Appendix D.1; full hyperparameters in Appendix E). The resulting reasoner produces well-structured analyses but remains generator-agnostic: it searches for anything that *might* be difficult, not only what *one specific* generator actually fails on.

Phase 1: Teach what the generator can internalize. The warmed-up reasoner provides search-augmented inputs (visual references and text prompts), exposing the generator to knowledge it previously lacked. Based on original text prompts and augmented inputs, we train generators to internalize world knowledge via online iterative Diffusion-DPO [32, 37]. In each episode, the generator samples M images per prompt, scored by the SEARCHGEN-20K evaluation protocol, and constructs preference pairs from top- and worst-scored generations. DPO reinforces the generator's own best outputs, internalizing knowledge whose visual identity is stable enough to absorb into parameters. After Phase 1, the generator $G_\theta^{(1)}$ has expanded its knowledge boundary: concepts that previously required search now reside in parameters (Figure 8b provides direct evidence: the CDF of per-prompt no-search quality shifts rightward after DPO).

DPO training also builds noise-robustness in visual generation: because the generator trains on search-augmented inputs, it learns to incorporate imperfect visual references without being dominated by their noise. A generator trained only on clean inputs treats every conditioning signal as authoritative, the root cause of concept corruption and copy effects identified in §2.5. Exposure to the noisy, varied outputs of real search builds robustness to imperfect references. This noise resistance is essential for Phase 2, because even a well-calibrated reasoner cannot guarantee perfectly clean references at inference time.

Phase 2: Search what the generator cannot be taught. The generator's boundary has shifted; the reasoner's policy is now miscalibrated, still searching for concepts the generator has internalized.

Recalibration uses rejection-sampling finetuning (RFT): we roll out N trajectories from the Phase 0 reasoner paired with $G_\theta^{(1)}$, score the resulting images, compute group-relative advantages, and retain only positive-advantage trajectories for continued reasoner training. Trajectories where the reasoner correctly abstains receive high scores and are reinforced; trajectories where unnecessary search corrupts output receive low scores and are discarded. The reasoner learns the strengthened generator's boundary without explicit boundary labels: the boundary emerges from the reward signal. We roll out the search agent's trace, score the resulting images, and reinforce positive-advantage trajectories, optimizing the agent's policy against the generator as its environment. The full RFT cycle fits in 4×8 GPU hours.

Taken together, the two phases execute complementary movements on the knowledge boundary: Phase 1 moves the boundary outward by internalizing stable knowledge and building noise robustness; Phase 2 moves the search policy inward to match, so that search fires precisely where knowledge remains outside the generator's expanded capacity.

Algorithm 1 Co-Training: Teach, Then Search

Require: Generator G_θ, base VLM R_ϕ, dataset $\mathcal{D} = \{p_i\}_{i=1}^{N_\mathrm{D}}$, quality function Q, margin ϵ
Ensure: Co-trained generator $G_{\theta'}$, calibrated reasoner $R_{\phi'}$

— Phase 0: Supervised Warm-Start —

Collect expert trajectories $\mathcal{T}_\mathrm{sft} = \{(p_i, a_i^\mathrm{A}, a_i^\mathrm{B}, a_i^\mathrm{C})\}$ for Tasks A/B/C ▷ ∼20K
$R_{\phi_0} \leftarrow \arg\min_\phi \sum_{(p,a)\in\mathcal{T}_\mathrm{sft}} \mathcal{L}_\mathrm{SFT}(\phi; p, a)$ ▷ Cross-entropy on structured task outputs

— Phase 1: Online Iterative DPO (Generator) —

for each episode $e = 1, \ldots, T_1$ **do**
  Sample batch $\mathcal{B}_e \subset \mathcal{D}$
  for each $p \in \mathcal{B}_e$ **do**
    $\tilde{p} \leftarrow R_{\phi_0}(p)$ ▷ Search-enriched specification via gate–filter–integrate
    Generate M images: $\{x_j\}_{j=1}^M \sim G_\theta(\cdot \mid \tilde{p})$
    Score: $s_j \leftarrow \mathrm{Score}(p, x_j)$ for each j ▷ Image scoring function; distinct from boundary-defining Q in Def. 1
    Construct preference pair: $x^w \leftarrow x_{\arg\max s_j}$, $x^l \leftarrow x_{\arg\min s_j}$
  end for
  Update via DPO loss adapted for flow-matching:
  $$\mathcal{L}_\mathrm{DPO}(\theta) = -\mathbb{E}\Big[\log\sigma\Big(\beta\left(\log\frac{v_\theta(x_t^w|\tilde{p})}{v_{\theta_\mathrm{ref}}(x_t^w|\tilde{p})} - \log\frac{v_\theta(x_t^l|\tilde{p})}{v_{\theta_\mathrm{ref}}(x_t^l|\tilde{p})}\right)\Big)\Big]$$
  where v_θ is the flow-matching velocity field, t is the diffusion timestep, $\beta = 100$ is the DPO temperature, and θ_ref is updated via EMA (decay 0.99). Preference pairs are constructed from groups of M generations per prompt.
  $\theta \leftarrow \theta - \eta_1 \nabla_\theta \mathcal{L}_\mathrm{DPO}(\theta)$
end for
$G_{\theta'} \leftarrow G_\theta$ ▷ DPO-strengthened generator

— Phase 2: Rejection Finetuning (Reasoner) —

for each $p \in \mathcal{D}_\mathrm{rft} \subset \mathcal{D}$ **do**
  Roll out N_traj trajectories: $\{\tau_n\}_{n=1}^{N_\mathrm{traj}}$, where $\tau_n = R_{\phi_0}(p)$, each yielding image $x_n \sim G_{\theta'}(\cdot \mid \tau_n)$
  Score: $s_n \leftarrow \mathrm{Score}(p, x_n)$
  Compute group-relative advantage: $A_n \leftarrow \frac{s_n - \bar{s}}{\sigma_s + \delta}$, $\bar{s} = \frac{1}{N_\mathrm{traj}}\sum_n s_n$, $\sigma_s = \mathrm{std}(\{s_n\})$
end for
Filter: $\mathcal{T}_\mathrm{rft} \leftarrow \{\tau_n : A_n > 0\}$ ▷ Retain positive-advantage trajectories
$R_{\phi'} \leftarrow \arg\min_\phi \sum_{\tau\in\mathcal{T}_\mathrm{rft}} \mathcal{L}_\mathrm{SFT}(\phi; \tau)$ ▷ SFT on filtered trajectories

return $(G_{\theta'},\ R_{\phi'})$

4 Validating Evolving Knowledge Boundary through Co-Training

If the knowledge boundary is real and co-training discovers it, three consequences follow. First, each training phase should improve overall quality without regressing on any stratum; *monotonicity* rules out the possibility that gains on search-intensive prompts come at the cost of prompts the generator already handles. Second, the improvement should be *selective*: the calibrated reasoner must recover performance on prompts that generic search degrades, demonstrating that it has learned *when not to search*. Third, the optimal search policy should be *generator-specific*: a policy calibrated to the base generator should behave differently from one calibrated to the DPO-strengthened generator, confirming that the boundary is a joint property of the generator–reasoner pair rather than a fixed property of the prompt distribution. We test each prediction below; per-dimension breakdowns and the broader generator landscape appear in the appendix.

Generators. Co-training experiments use two architecturally distinct open-weight generators: Flux.2-Klein-4B (**Klein-4B**), a flow-matching model, and Bagel-7B (**Bagel**), a unified vision–language model. This model choice is deliberate because the two models are widely acknowledged open-source generators that support both text-to-image generation and reference-to-image generation. Testing on two architectures rules out the possibility that co-training benefits are specific to one conditioning mechanism. Each has a DPO-finetuned variant (**Klein-4B-DPO** and **Bagel-DPO**) whose knowledge boundary has shifted outward through Phase 1 training.

Reasoner variants. The progression from weaker to stronger reasoners isolates the effect of generator-aware calibration: **NO SEARCH**: generator alone (no retrieval baseline). **BLIND SEARCH** (SFT-8B): Qwen3-VL-8B after supervised warm-up; the reasoner follows the protocol to perform reasoning and search but is generator-agnostic, i.e., not aware of knowledge boundary of specific generators. **GENERATOR-ADAPTIVE SEARCH**: undergoes co-training based on BLIND SEARCH; the reasoner is calibrated to specific generator's specific boundary. **ORACLE (Frontier API)**: trillion parameters high-cost Gemini-3-Flash API as the reasoner following same reasoning and search protocol. This serves as an upper bound on generator-agnostic reasoner, showing best possible performance when pairing generators with strong but generic reasoner.

All reasoners are equipped with search tools for world-knowledge grounding. We use Google Image and Web Search via SERP API.

Evaluation. All scores use the Gemini-3-Flash judge on the 751-prompt test set (0–100 scale, 9-component rubric averaged to an overall score; Appendix D.2). The search-intensive prompts are partitioned into three difficulty sets by Nano Banana Pro no-search quality (tercile boundaries): Set I (easiest), Set II, and Set III (hardest).

4.1 Co-Training Produces Monotonic, Selective Improvement

With the setup in place, we first test our key insight: that co-training *discovers and expands* the generator's knowledge boundary. Figure 8 supplies the two most direct pieces of visual evidence. Panel (a) shows the co-training progression—Reasoner SFT, Generator DPO, Reasoner RFT—improving monotonically within every difficulty tier, previewing the downstream quality gains we quantify below. Panel (b) exposes the underlying mechanism: the CDF of per-prompt no-search quality for the base vs. DPO-strengthened generator shifts rightward, meaning fewer prompts score low and more score high from parametric knowledge alone. The shaded region is newly internalized knowledge—concepts whose per-prompt quality gap formerly exceeded ϵ and now fall below it, migrating from $\mathcal{K}_{\text{ctx}}$ to $\mathcal{K}_{\text{int}}$ (Def. 1). The shift holds across both Klein-4B and Bagel-7B, confirming that DPO expands the boundary regardless of architecture.

> **Finding 3: Co-training discovers the knowledge boundary: a calibrated 8B reasoner matches a frontier oracle on the same generator.** The monotonic progression from NO SEARCH through BLIND SEARCH to GENERATOR-ADAPTIVE SEARCH, the per-stratum recovery on NoSearch prompts, and the generator-specific behavior across two architectures provide converging evidence that the structural split is an operational design variable, not merely a conceptual framework (Table 6).

Table 6 quantifies this progression. The top section traces three phases in order: Phase 0 pairs a generator-agnostic SFT reasoner with the base generator; Phase 1 keeps the same reasoner but pairs it

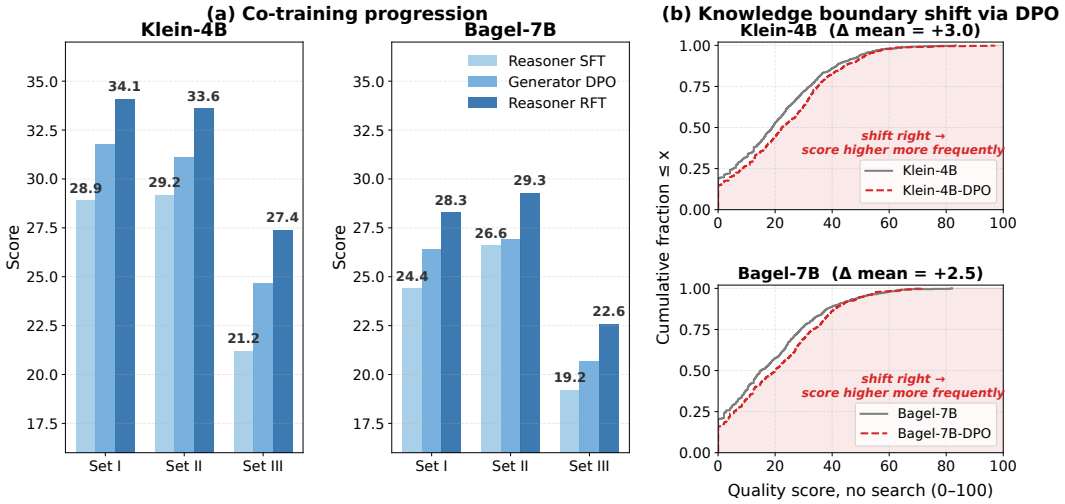

Figure 8: **Co-training progression and knowledge boundary shift. (a)** Grouped bars show the three co-training stages (Reasoner SFT, Generator DPO, Reasoner RFT) for Set I (easiest), Set II, and Set III (hardest) search-intensive tiers (Klein-4B). All three tiers show monotonic improvement. **(b)** Cumulative distribution of per-prompt no-search quality scores for base vs. DPO-finetuned generators on the 647-prompt eval set. The DPO curve sits below the base curve, indicating a rightward shift: fewer prompts receive low scores and more receive high scores from parametric knowledge alone. The shaded region represents newly internalized knowledge: concepts that previously required search but now reside in generator parameters. The shift is consistent across both Klein-4B (top) and Bagel-7B (bottom), confirming that DPO expands the knowledge boundary regardless of architecture.

Table 6: **Main results: the co-training progression.** The **Progression** section traces the three co-training phases for both Klein-4B and Bagel. The **Reference baselines** section reports the no-search floor, frontier oracle ceiling, and cross-generator check for each architecture. Difficulty terciles (Set I–III) by Nano Banana Pro no-search quality; NoSearch subset (100 prompts) reported separately.

		NoSearch	Search			
			Set I	Set II	Set III	Overall
Co-training progression						
Klein-4B						
Phase 0	BLIND SEARCH (SFT-8B) + Klein-4B	54.6	28.9	29.2	21.2	26.4
Phase 1	BLIND SEARCH (SFT-8B) + Klein-4B-DPO	54.0	31.8	31.1	24.7	29.2
Phase 2	GENERATOR-ADAPTIVE SEARCH (RFT-8B) + Klein-4B-DPO	**56.9**	**34.1**	**33.6**	**27.4**	**31.8**
Bagel						
Phase 0	BLIND SEARCH (SFT-8B) + Bagel	52.6	24.4	26.6	19.2	23.4
Phase 1	BLIND SEARCH (SFT-8B) + Bagel-DPO	52.4	26.4	26.9	20.7	24.7
Phase 2	GENERATOR-ADAPTIVE SEARCH (RFT-8B) + Bagel-DPO	**54.3**	**28.3**	**29.3**	**22.6**	**26.8**
Reference baselines						
Klein-4B						
	NO SEARCH + Klein-4B-DPO	49.9	28.2	26.3	20.6	25.0
	ORACLE + Klein-4B-DPO	55.7	33.7	33.9	26.0	31.2
	GENERATOR-ADAPTIVE SEARCH (RFT-8B) + Klein-4B	54.6	29.0	29.8	21.5	26.8
Bagel						
	NO SEARCH + Bagel-DPO	48.8	24.3	24.0	18.8	22.4
	ORACLE + Bagel-DPO	53.2	28.1	29.0	21.1	26.1
	GENERATOR-ADAPTIVE SEARCH (RFT-8B) + Bagel	52.5	25.1	27.3	19.8	24.1

with the DPO-strengthened generator (isolating the generator improvement); Phase 2 recalibrates the reasoner to the strengthened generator via RFT (isolating the reasoner calibration effect). The bottom section provides reference baselines that bound interpretation: no-search baselines establish the floor, frontier oracle baselines establish the ceiling, and a cross-check (calibrated reasoner on base generator) tests generator-specificity directly.

Three patterns emerge, each confirming a prediction of the co-training principle.

Monotonicity. The progression is monotonic on both generators. On Klein-4B, Phase 2 reaches 31.8, slightly exceeding the frontier oracle on the same generator (31.2). Both training phases contribute independently: generator DPO alone (Phase 0→1) yields +2.8; reasoner RFT (Phase 1→2) adds +2.6. The same pattern holds for Bagel (Table 6).

As Figure 8(a) confirms, the improvement is monotonic within every difficulty tier, with Set III (hardest) showing the largest gains where the boundary shift (Figure 8b) has the most room to operate.

Selectivity. Recall from §2.5 that naive search *degrades* prompts the generator already handles (Finding 2). The calibrated reasoner must therefore improve search-intensive prompts without sacrificing quality on prompts where search is harmful. The NoSearch column in Table 6 tests this directly: Phase 2 scores 56.9, a +7.0 gain over the no-search DPO baseline (49.9). The reasoner has learned when to abstain; it no longer corrupts prompts the strengthened generator already handles. Within the Search partition, Phase 2 improves over the no-search DPO baseline at every set and matches or closely approaches the frontier oracle, with the largest gains on Set III where external knowledge is most critical. The same selectivity pattern holds for Bagel (Phase 2: 54.3 on NoSearch vs. 48.8 for no-search DPO; 26.8 overall).

Generator-specificity. If the knowledge boundary is generator-specific, a reasoner calibrated to the DPO-strengthened generator should behave differently when paired with the base generator, because the base generator's boundary sits in a different location. The cross-check rows in Table 6 test this: GENERATOR-ADAPTIVE SEARCH paired with *base* Klein-4B scores 26.8 overall (vs. 31.8 when paired with Klein-4B-DPO, the generator it was calibrated for). NoSearch recovery remains strong (the reasoner still knows when to abstain), but Set III gains shrink because the reasoner searches for concepts the base generator cannot render even with correct references. The calibrated reasoner, when paired with its *intended* generator, matches or exceeds frontier-oracle scores at every tier. This asymmetry confirms that the search policy is a joint property of the generator–reasoner pair: a policy optimal for one generator is suboptimal for another.

Structural interpretation. The evidence confirms all three predictions: monotonicity across phases and generators, selectivity on the NoSearch stratum, and generator-specificity in the cross-check. The matched-compute comparison is the sharpest validation: our co-trained 8B reasoner paired with a 4B generator (31.8) slightly exceeds a frontier VLM oracle on the same generator (31.2), suggesting that generator-specific calibration can approach costly frontier-scale reasoning at a fraction of the compute, latency and expense. The 39-point gap to GPT-Image-2 (71.0) reflects generator capacity at the 4B scale rather than a framework limitation; the co-training principle extracts the maximum from a fixed generator, and scaling the generator along this axis is the clear path to closing the absolute gap.

5 Related Work

Search-augmented generation. Retrieval-augmented generation (RAG) [26] and REALM [16] established the paradigm of grounding language models in external documents at inference time. Atlas [20] and REPLUG [35] take this further by jointly training the searcher and language model end-to-end, demonstrating that aligning the retrieval distribution with the generator's needs yields substantial gains over frozen-searcher pipelines. In the visual domain, Re-Imagen [9] conditions diffusion on searched image–text pairs, RDM [6] augments generation with nearest-neighbor features, KNN-Diffusion [2] queries a large image database at generation time, and IP-Adapter [49] injects visual exemplars through cross-attention adapters. Subject-driven methods such as DreamBooth [33] and Textual Inversion [14] internalize specific concepts via per-concept optimization; BLIP-Diffusion [28] pre-trains a subject representation for efficient control. A common thread unites these pipelines: search is *always on*, triggering retrieval for every prompt regardless of whether the generator already possesses adequate knowledge. Our analysis (§2) shows that this unconditional policy actively degrades a substantial fraction of prompts, establishing that the *decision to search* is itself a first-order design variable.

Selective and adaptive retrieval. The question of *when* to search has been studied extensively in text. Self-RAG [1] trains a language model to emit retrieval-control and critique tokens. FLARE [23] triggers retrieval when predictive confidence drops. Toolformer [34] learns when to invoke external APIs including search. CRAG [47] evaluates and corrects retrieval quality post-hoc. Adaptive-RAG [21] routes queries to different retrieval strategies based on complexity. Mallen et al. [30]

and Kandpal et al. [24] show that popular entities are better served by parametric knowledge while rare entities benefit from retrieval, an empirical split analogous to our internalizable/contextual distinction. These methods establish selective retrieval for text, but visual search failure is structurally different: it manifests as spatial distortion, identity blending, and stylistic contamination (§2.5), failure modes that have no text-domain analogue and require modality-specific triage and integration. More fundamentally, none co-trains the search policy with the generator: the retrieval decision is learned against a fixed generation model, whereas our co-training framework recalibrates the search policy after each generator improvement.

Tool-augmented and agentic image generation. GenAgent [22], RPG-DiffusionMaster [48], LLM-grounded Diffusion [29], and IterComp [51] use LLM-based planning to decompose complex prompts, improving compositional generation within the generator's existing knowledge. Instruct-Imagen [17] supports multi-modal instructions including style exemplars. RationalRewards [42] show that preference-calibrated prompt rewriter significantly improve image generation via agentic workflow at test-time. These systems assume the generator possesses the relevant visual concepts and focus on compositional arrangement; none addresses knowledge *absence*, the case where the generator has never seen the concept it must render. Our work is complementary: agentic planning can improve the compositional quality of search-augmented outputs, but cannot substitute for the missing knowledge that search provides.

Knowledge-intensive evaluation. Existing benchmarks evaluate compositionality and aesthetic quality within known visual concepts: GenAI-Bench [27] tests prompt adherence, T2I-CompBench [19] evaluates spatial reasoning and attribute binding, DALL-Eval [11] measures object relationships, and HEIM [25] provides holistic multi-dimensional evaluation. VLM-based automated judges such as TIFA [18] and human-grounded auto-metric suites such as Gecko [45] enable scalable faithfulness assessment. A prompt like "a red cube on a blue sphere" tests spatial reasoning but assumes the generator knows what cubes and spheres look like. SEARCHGEN-BENCH is complementary: it targets concepts *outside* the training distribution (cultural symbols, recent entities, niche typography), measuring knowledge absence rather than reasoning failure. The gap is stark: Flux.2-Klein-9B scores above 4.0/5.0 on GenAI-Bench yet falls below 30/100 on SEARCHGEN-BENCH, a discrepancy of over 60 percentage points that no existing evaluation captures.

Self-improving and co-training systems. Self-Rewarding Language Models [50] show that a model can judge its own outputs to generate training signal, improving over successive iterations without external reward. SPIN [10] frames self-improvement as self-play, iteratively distinguishing model outputs from ground truth. Our co-training framework instantiates this principle for visual generation: Phase 1 (DPO) teaches the generator from search-augmented demonstrations that expose knowledge it previously lacked, while Phase 2 (RFT) recalibrates the reasoner to the strengthened generator's shifted boundary. The deliberate ordering (teach, then search) and the boundary-discovery mechanism distinguish our approach from generic self-play, where both players improve symmetrically; here the generator and reasoner occupy structurally different roles and improve in complementary directions.

6 Conclusion

The central question is not how to build a model that knows everything, but how to build a system that knows what it does not know. This split creates a generator-specific knowledge boundary, and we have shown that this boundary, though hard to specify a priori, is discoverable through co-training. Our evidence suggests that this self-awareness emerges from co-training dynamics, not from scale alone.

The resolution lies in co-training the generator and reasoner around their shared boundary. Online DPO serves a dual function: it pushes the generator's boundary outward by internalizing stable visual knowledge, and it teaches the generator to use imperfect references without being corrupted by them. Rejection finetuning then recalibrates the reasoner to search only what the strengthened generator still cannot render. The monotonic progression from NO SEARCH through BLIND SEARCH to GENERATOR-ADAPTIVE SEARCH, the per-stratum recovery on NoSearch prompts, and the generator-specific behavior across two architectures provide converging evidence that the structural split is an operational design variable, not merely a conceptual framework.

Our deliberately minimal recipe (one DPO pass, one RFT pass, a 4B generator, an 8B reasoner) validates the principle at the smallest useful scale. Each subsequent iteration can push the boundary

further outward and tighten the search policy further inward, converging toward a regime where only genuinely contextual knowledge triggers search. A natural objection is that next-generation models will absorb more knowledge at scale, making external search unnecessary. But training data is finite while the world is unbounded and in constant motion: no model, regardless of scale, can internalize knowledge about events after its training cutoff, entities too rare for any feasible dataset, or cultural knowledge that evolves across communities and time. The knowledge boundary may shift outward with scale, but it cannot disappear; co-training discovers where it lies for any given generator, at any scale. Whether additional iterations sharpen the boundary, whether larger generators internalize knowledge that resists parameterization in smaller ones, and whether richer reward signals accelerate gains on the hardest strata are open questions that define a clear research trajectory.

Beyond these immediate questions, this work opens several broader research directions. What is the scaling law of the knowledge boundary: does the internalizable set expand uniformly across failure categories as generator size increases, or do certain categories (e.g., cultural specificity) resist internalization? Can the boundary be predicted from model internals without full co-training, for instance via probing classifiers or uncertainty estimates? Does the internalizable/contextual split generalize to other modalities such as video, 3D, and music generation, which face the same structural problem of finite training data and unbounded user requests?

Beyond search: the knowledge boundary as a tool-invocation principle. Search is the first and most fundamental tool, but the boundary that governs *when to invoke it* is not search-specific. The same gate–filter–integrate protocol and teach-then-search co-training extend to other tools—image editing and inpainting, render-as-code (SVG, matplotlib, CAD), 3D-asset retrieval, and structural control—each supplying a different slice of the contextual knowledge set $\mathcal{K}_{\text{ctx}}$. Whether a single agent can learn a unified boundary across such a toolbox is an open direction this work's harness is built to support.

The released dataset, co-training corpus, and search corpus provide the infrastructure for pursuing these questions: 20,839 prompts, 90,452 reasoning trajectories, 281,925 generated images, and 145,642 archived search sessions that can be replayed without live search API access.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5goOZv.

[2] Oron Ashual and Lior Wolf. KNN-diffusion: Image generation via large-scale retrieval. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1b3Ndz5iUf.

[3] Jason Baldridge, Tim Brooks, Trevor Darrell, Jeff Dean, Patrick Esser, Deep Ganguli, Amir Ghiasi, Tom Henighan, Daniel Ho, Timo Höppe, et al. Imagen 3. Technical report, Google DeepMind, 2024.

[4] Black Forest Labs. FLUX.2: Frontier visual intelligence. Model release and technical documentation, 2026. URL https://bfl.ai/. We evaluate the open-weight FLUX.2 [klein] 4B and 9B text-to-image checkpoints.

[5] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Patrick Esser, and Björn Ommer. Retrieval-augmented diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 10671–10684, 2022.

[7] ByteDance. Jimeng AI: Intelligent image creation. Product documentation, 2025. URL https://jimeng.jianying.com/. Commercial text-to-image endpoint (`jimeng`) used in generator landscape experiments.

[8] ByteDance Seed Team. Seedream 4.0: Toward next-generation multimodal image generation, 2025. URL https://arxiv.org/abs/2509.20427.

[9] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. Re-Imagen: Retrieval-augmented text-to-image generator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XSEBx0iSjFQ.

[10] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[11] Jaemin Cho, Abhay Zala, and Mohit Bansal. DALL-Eval: Probing the reasoning skills and social biases of text-to-image generation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3043–3054, 2023.

[12] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

[14] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] Google DeepMind. Gemini image generation (gemini 2.5 Flash / gemini 3 Pro). Product and API documentation, 2025. URL https://ai.google.dev/gemini-api/docs. Consumer-facing names Nano Banana and Nano Banana Pro refer to Gemini 2.5 Flash Image and Gemini 3 Pro Image endpoints accessed via the Gemini developer API in our experiments.

[16] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020.

[17] Hexiang Hu, Kelvin C. K. Chan, Yu-Chuan Su, Wenhu Chen, Yandong Li, Kihyuk Sohn, Yang Zhao, Xue Ben, Boqing Gong, William W. Cohen, Ming-Wei Chang, and Xuhui Jia. Instruct-Imagen: Image generation with multi-modal instruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4754–4763, 2024.

[18] Yushi Hu, Baolin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Hannaneh Hajishirzi. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20411–20432, 2023.

[19] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[20] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. In *Journal of Machine Learning Research*, volume 24, pages 1–43, 2023.

[21] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*, 2024.

[22] Kaixun Jiang, Yuzheng Wang, Junjie Zhou, Pandeng Li, Zhihang Liu, Chen-Wei Xie, Zhaoyu Chen, Yun Zheng, and Wenqiang Zhang. GenAgent: Scaling text-to-image generation via agentic multimodal reasoning, 2026. URL https://arxiv.org/abs/2601.18543.

[23] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992. Association for Computational Linguistics, 2023.

[24] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR, 2023.

[25] Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, Minguk Kang, Taesung Park, Jure Leskovec, Jun-Yan Zhu, Li Fei-Fei, Jiajun Wu, Stefano Ermon, and Percy Liang. Holistic evaluation of text-to-image models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[27] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. GenAI-Bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024. URL https://arxiv.org/abs/2406.13743.

[28] Dongxu Li, Junnan Li, and Steven C. H. Hoi. BLIP-Diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[29] Long Lian, Boyi Li, Anima Anandkumar, Eric P. Xing, and Jiayuan Li. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, 2023.

[31] OpenAI. GPT Image: Openai image generation. Product and API documentation, 2025. URL https://platform.openai.com/docs/guides/image-generation. We refer to the OpenAI image-generation endpoint as GPT-Image-2 in our experiments.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine-tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[34] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[35] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

[36] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[37] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8387–8396, 2024.

[38] Haozhe Wang, Long Li, Chao Qu, Weidi Xu, Fengming Zhu, Wei Chu, and Fangzhen Lin. To code or not to code? adaptive tool integration for math language models via expectation-maximization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3060–3075, 2025.

[39] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight.

[40] Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[41] Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, et al. Reverse-engineered reasoning for open-ended generation. In *International Conference on Learning Representations (ICLR)*, 2026.

[42] Haozhe Wang, Cong Wei, Weiming Ren, Jiaming Liu, Fangzhen Lin, and Wenhu Chen. Rationalrewards: Reasoning rewards scale visual generation both training and test time. *arXiv preprint arXiv:2604.11626*, 2026.

[43] Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in llms through reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

[44] Haozhe Wang, Qixin Xu, Changpeng Wang, Taofeng Xue, Chong Peng, Wenhu Chen, and Fangzhen Lin. Bad seeing or bad thinking? rewarding perception for multimodal reasoning, 2026. URL https://arxiv.org/abs/2605.14054.

[45] Olivia Wiles, Chuhan Zhang, Isabela Albuquerque, Ivana Kajic, Su Wang, Emanuele Bugliarello, Yasumasa Onoe, Chris Knutsen, Cyrus Rashtchian, Jordi Pont-Tuset, and Aida Nematzadeh. Revisiting text-to-image evaluation with Gecko: On metrics, prompts, and human ratings. *arXiv preprint arXiv:2404.16820*, 2024.

[46] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, Zenan Liu, et al. Qwen-Image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

[47] Shiqi Yan, Jiachen Gu, Yun Zhu, and Zhenhua Ling. CRAG: Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

[48] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56704–56721. PMLR, 2024. URL https://proceedings.mlr.press/v235/yang24ai.html.

[49] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

[50] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[51] Xinchen Zhang, Ling Yang, Guohao Li, Yaqi Cai, Jiake Xie, Yong Tang, Yujiu Yang, Mengdi Wang, and Bin Cui. IterComp: Iterative composition-aware feedback learning from model gallery for text-to-image generation. *arXiv preprint arXiv:2410.07171*, 2024.

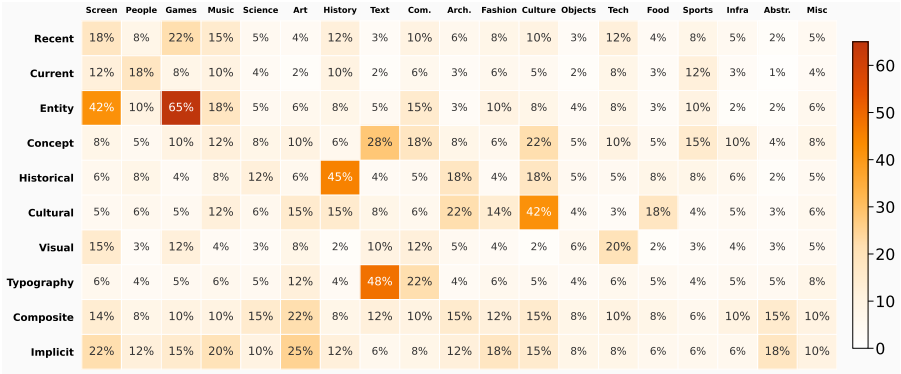

Figure 9: **The bottleneck is pervasive: every domain–category combination shows knowledge-driven degradation.** Heatmap summarizing cross-category structure between failure modes and domain categories in SEARCHGEN-20K. The uniform severity across all cells rules out the hypothesis that failures concentrate in a few niche categories.

Appendix overview

This appendix is organized to follow the evidence chain of the paper. §A.1 documents benchmark construction and the per-category/domain heatmap used by the main text. §B.1–§B.2 provide focused failure-analysis context and judge–human agreement. §C.1 gives the full nine-component score breakdown across all generators. §D.1–§D.2 provide the end-to-end workflow, reasoner I/O examples, and evaluation protocol. §E documents hardware and training/serving implementation details. §F–§G summarize limitations, release safeguards, third-party/API caveats, and broader impact.

A Benchmark and dataset details

A.1 Benchmark construction details

Taxonomy annotation reliability. Two annotators independently labeled 2,000 sampled failure cases, iteratively refined the schema to Cohen's $\kappa > 0.85$, then applied it to all 10,840 prompts in the production scrape. The taxonomy's stability across annotators supports viewing it as capturing structure in the failure space rather than an artifact of labeling choices.

The failure taxonomy appears in Table 1 (main paper).

Layer 1 (seed entities). From the 10,840 real user prompts, we extract a seed database of 31,537 entities spanning 22 primary domains, each annotated with canonical name, estimated training-set frequency, ground-truth visual references, and distinguishing attributes. The distribution is intentionally long-tailed, mirroring real user demand.

Layer 2 (prompt synthesis). From the seed database, we synthesize the SEARCHGEN-20K corpus in three steps. First, *template instantiation* produces structured drafts across categories and domains by filling parameterized slots for entities, relations, and stylistic constraints distilled from the production scrape. Second, a *frontier large language model* rewrites each instantiated template into a naturalistic user-style request while preserving grounded entities, modality requirements (image versus text retrieval), and checklist-eligible facts; temperature is elevated to diversify surface form and reduce templating artifacts. Third, *quality control* removes prompts that are ambiguous under the evaluation protocol, culturally insensitive, or misaligned with the intended failure mode; annotators also verify that checklist items remain answerable from the prompt and cached search evidence alone. Each accepted prompt includes 2–4 ground-truth search queries, a structured list of 3–8 key visual elements with binary verification checklists, and a 1–5 evaluation rubric tailored to the failure category. This annotation infrastructure enables fully automated evaluation via a Gemini-3-Flash VLM judge [15] (Spearman $\rho = 0.87$ with human ratings; methodology in Appendix B.2).

Cultural diversity and splits. The benchmark spans global cultural diversity—entities from East Asia, South Asia, the Middle East, Africa, Latin America, and beyond—because knowledge gaps concentrate in long-tail cultural concepts and a Western-centric benchmark would underestimate

severity. For supervised training and model selection we use train (20,000), validation (128), and **test** (751) partitions, matching Section 2.2; remaining non-test prompts support cached search construction, trajectory collection, and co-training rollouts. Unlike benchmarks that test compositionality within known concepts [27, 19, 11], SEARCHGEN-BENCH targets knowledge absence.

B Additional empirical analysis

B.1 Extended failure analysis

Concept corruption (gating). The generator's conditioning architecture typically has no reliable mechanism for gating external information based on its own confidence in its internal representation. When a reference image (or strongly textually specified exemplar) is provided, it is integrated with substantial weight regardless of whether the generator's native rendering would have been more accurate on the same entity, yielding concept corruption on prompts where internal knowledge was already adequate.

Reference copying (pixel-level conditioning). Visual conditioning operates on pixel (and low-level feature) statistics rather than explicit semantic attributes. Without an integration layer that specifies which attributes to transplant, the generator tends to reproduce incidental structure from the reference—background, lighting, framing—so the output resembles a filtered search result rather than a new composite generation.

Why surface cues are a poor proxy for Qwen-Image-boundary strata. Named entities and other shallow triggers are only loosely correlated with the knowledge boundary inferred from Qwen-Image in Table 4: many prompts contain proper nouns that already lie inside Qwen-Image's *NoSearch* stratum, while some high-gap prompts yield few detectable entities. Cheap gates therefore trigger search on the wrong subset, repeating the same tradeoff between *VisualSearch*/*TextualSearch* gains and *NoSearch* regressions that *BlindSearch* exhibits outright.

TextualSearch stratum (extended discussion). The main paper (§4.1) compresses the TextualSearch finding: for Klein-4B-DPO, both BLIND SEARCHand GENERATOR-ADAPTIVE SEARCHscore below the no-search baseline because internalized contextual knowledge makes search net harmful on the hardest prompts. Those prompts disproportionately require compositional contextual reasoning—typography, cultural composition, and implicit interpretation—that resists both single-pass parameterization and single-pass search, motivating richer co-training cycles and search strategies.

B.2 Judge–human correlation and judge–reasoner independence

Human study protocol. Human ratings are collected on the same 500 prompt–image pairs used to report aggregate agreement in §2.2. Each pair is presented with the user prompt, the checklist, and the category rubric; raters assign checklist judgments, rubric scores, and holistic quality judgments using the same anchors as the automated judge. Multiple raters participate with adjudication on high-disagreement items so that gold labels are stable under resampling.

Aggregate agreement. The Gemini-3-Flash judge achieves Spearman $\rho = 0.87$ with the consolidated human labels on this set, which is consistent with scalable VLM-as-judge practice for text-to-image faithfulness evaluation [18, 45, 25].

Stratified difficulty. Agreement is tightest on object-centric *NoSearch* prompts where errors are predominantly omissions of salient attributes. Agreement weakens on *TextualSearch*, where glyph-level correctness and fine layout dominate rubric variance; this pattern aligns with known limitations of VLMs on dense in-image text and motivates treating TextualSearch as a conservative stratum in automated evaluation.

C Extended quantitative results

C.1 Full nine-component breakdown by stratum

Table 7 provides the complete nine-component evaluation for all generators on both the NoSearch and Search-Intensive strata. This supplements the six-component summary in the main paper (Table 4) with the full scoring breakdown.

Table 7: **The Search Bottleneck on SEARCHGEN-BENCH – Full Nine-Component Breakdown.** Scores on a 0–100 scale (one decimal); higher is better. Every generator drops sharply on the *Search-Intensive* set, confirming that the bottleneck reflects missing knowledge rather than rendering ability. Knowledge-sensitive components (Checklist, Rubric, Visual Ref.) vary per prompt and measure knowledge presence; knowledge-invariant components (Text Rend., Phys. Plaus., Img. Qual.) measure rendering competence.

Type	Generator	Overall	Checklist	Rubric	Prompt	Img. Qual.	Text Rend.	AI Nat.	Comp. Aes.	Phys. Plaus.	Vis. Ref.
NoSearch set (parametric knowledge sufficient)											
Open	Bagel	49.3	52.3	49.1	39.2	58.0	18.4	48.3	65.3	60.4	34.8
	Flux.2-Klein-4B	51.0	54.7	51.8	39.3	61.3	13.2	49.5	67.2	63.5	32.6
	Flux.2-Klein-9B	57.8	63.8	59.8	51.3	63.8	31.3	52.5	72.3	66.7	43.7
	Qwen-Image	67.4	74.8	70.8	62.8	68.3	63.0	61.0	76.8	73.8	56.7
Commercial	Qwen-Image-2	70.7	78.7	73.8	68.4	70.2	71.7	60.8	80.3	75.6	60.0
	SeedDream-4.0	67.9	74.9	70.6	61.7	69.5	71.1	59.7	80.3	73.9	56.7
	Nano Banana	63.1	71.1	67.5	59.7	66.3	42.8	56.5	76.3	68.5	53.8
	Nano Banana Pro	75.0	82.8	78.1	72.8	71.5	85.9	65.0	83.3	78.2	67.8
	GPT-Image-2	71.1	78.6	75.6	70.8	69.0	67.7	60.8	77.7	73.9	64.3
Search-Intensive (external knowledge required)											
Open	Bagel	21.5	18.2	17.6	13.3	30.5	2.5	29.3	33.6	36.8	13.5
	Flux.2-Klein-4B	24.1	19.8	18.4	12.4	37.2	4.2	33.6	39.3	46.2	11.9
	Flux.2-Klein-9B	26.7	24.2	23.1	17.2	36.8	7.2	32.9	40.4	48.6	16.9
	Qwen-Image	27.9	24.8	24.3	18.6	40.1	8.7	31.6	42.8	44.6	17.7
Commercial	Qwen-Image-2	31.6	28.5	27.1	22.1	42.2	12.7	36.3	45.5	48.2	21.0
	Imagen3-Fast	14.1	9.7	10.0	6.9	22.2	1.4	21.4	23.2	23.7	7.0
	SeedDream-4.0	45.9	44.2	43.6	38.5	57.0	35.9	47.1	58.7	64.0	35.1
	Nano Banana	44.1	41.0	40.4	36.0	57.1	28.0	47.4	61.5	65.5	33.2
	Nano Banana Pro	65.3	64.4	63.1	60.7	71.4	65.0	62.0	75.9	78.5	58.3
	GPT-Image-2	71.0	71.2	70.1	69.2	75.1	75.9	64.7	80.4	77.3	66.0

D Protocols and supplementary artifacts

D.1 End-to-end workflow and reasoner I/O

The reasoner's three stages of §3.1—**gate** (Stage 1), **filter** (Stage 2), and **integrate** (Stage 3)—are stored in our SFT corpus as trajectory types Task A, Task B, and Task C, respectively. We first trace a *single* request end-to-end to show the overall inputs and outputs, then give one representative input–output pair per stage so the input contract and output fields (shown as `field`) are visible at a glance. We render outputs in compact form; the released co-training corpus contains the verbatim JSON in the exact schema. These are representative *successful* cases; failure modes such as copy effects are analyzed in §2.5.

End-to-end workflow (condensed). A user prompt enters; the reasoner runs gate $\rightarrow$ filter $\rightarrow$ integrate to decide what to search, select references, and emit an enriched prompt; the generator renders the image; and a VLM judge scores it against a checklist and rubric. The box below traces one request through these steps, and Figure 10 shows the retrieved references and the resulting generation.

End-to-end example (condensed).

Input — user request: *"Paint Yang Chaoyue on a 1970s Chinese rural field ridge; commune-member clothing; long-handled hoe; earth tones; 1970s film grain."*

Stage 1 (gate) — image search queries: Yang Chaoyue natural photo; 1970s commune labor scene; 1970s rural film photography; long-handled hoe.

Stage 2 (filter) — selected references: a period rural-ridge scene; commune work clothing; Yang Chaoyue likeness.

Stage 3 (integrate) — enriched prompt (output to generator): *"A 1970s Chinese rural labor scene on a yellow-earth field ridge: Yang Chaoyue in period commune clothing, holding a wooden long-handled hoe with a curved iron blade, natural expression, earth-tone palette, slight vintage color cast, film grain, no modern elements."*

Output — generation & evaluation: the enriched prompt and the three references condition the generator (Figure 10); a VLM judge then scores the result on likeness, historical authenticity, and atmospheric tone (protocol in §D.2).

Reference 1 (scene)

Reference 2 (costume)

Reference 3 (likeness)

Augmented output (Flux.2-Klein-4B-DPO).

Figure 10: **The end-to-end example's visual output.** The three references the reasoner retrieved and selected (top: scene, costume, likeness) and the image the generator produced from the enriched prompt (bottom), for the Yang Chaoyue request traced in the box above. This is the section's only image example; it grounds what "retrieved references" and "generation" concretely look like. The per-stage boxes that follow are text-only and use different prompts, each chosen to isolate one stage's behavior.

Stage 1 — Gate (SFT Task A). The gate decides *whether* external grounding is needed, classifies each surviving gap by type and severity, and proposes modality-labeled search queries. The two cases below show the decision it must make: it *fires* on a named landmark with a critical visual-identity gap, and it *skips* a generic studio prompt the base generator already renders reliably—the discipline that keeps the pipeline from over-retrieving.

Stage 1 (gate) — search fires.

Prompt: *"Photorealistic dusk poster of the Shanghai Tower showing its actual spiral facade and correct height proportions."*

Knowledge gap: Shanghai Tower — `category`: entity identity; `severity`: *critical*. The signature 120-degree spiral taper and tiered glass facade are frequently rendered incorrectly; an image reference fixes the exact geometry.

Decision (`search_decision`): *search*. **Query** (`modality`: image): "Shanghai Tower full exterior dusk photograph".

Justification: a critical visual-identity gap survives analysis, so image search is warranted.

Stage 1 (gate) — search skips.

Prompt: *"Photorealistic product shot of a blue and white porcelain teapot with bamboo motif on a marble tabletop, shallow depth of field, soft studio lighting."*

Knowledge gaps (`knowledge_gaps`): none. Porcelain teapots, bamboo motifs, marble surfaces, and studio lighting all render reliably from training data; no named entities, events, or cultural specifics are present.

Decision (`search_decision`): *skip*. No critical or important gap survives; searching would inject noise without improving accuracy.

Fields: `severity`$\in$\{critical, important, moderate, minimal\}; `modality`$\in$\{image, web\}.

Stage 2 — Filter (SFT Task B). Once the gate fires, retrieval returns several candidate references; the filter selects the single one that most directly fills the identified gap while minimizing extraneous content. Below, only one of three candidates carries the required location-specific layout, and the filter selects it by 0-based index.

Stage 2 (filter) — reference selection.

User request: *"Create a photorealistic render of the current Atlanta, GA UV index dashboard, showing both the real-time UV reading and the daily maximum forecast value."*

Search query: "Official Atlanta Georgia real-time UV index dashboard example".

Candidates (0-based):

0. generic Home Assistant UV-index card (dashboard widget)
1. Atlanta, GA real-time UV index dashboard
2. Buckhead dermatology clinic UV-safety page

Identified gap (`identified_knowledge_gaps`): the model lacks the location-specific layout, branding, and data structure of a real-time UV dashboard for Atlanta.

Selected (`selected_index`): **1**. Only candidate 1 shows an Atlanta-specific dashboard with the correct layout; candidates 0 and 2 are a generic widget and a clinical page and lack the required city-specific structure.

Stage 3 — Integrate (SFT Task C). The final stage folds the selected reference(s) into a generation-ready prompt, routing visual grounding *through language*: rather than passing raw pixels alone, the reasoner names exactly what to borrow with a grounded citation ("following Image 1, render ...") so the generator inherits the missing knowledge without copying unrequested style or layout.

Stage 3 (integrate) — reference-grounded refinement.

User request: *"Generate a photorealistic image of a polar bear standing in the Sahara Desert, looking confused as it searches for ice, with sand dunes and a single dead acacia tree nearby."*

Reference 1: a dead acacia tree under hard desert sunlight (retrieved for the tree's branching structure).

Borrow from Image 1 (`borrow_from_references`): the acacia tree's forked branching silhouette, dry cracked bark texture, and hard midday desert lighting.

Create new (`create_new`): the confused polar bear searching for ice; rolling Sahara dunes to the horizon; clear blue sky.

Refined prompt (`refined_prompt`): *"Photorealistic 8k polar bear in the Sahara, furrowed brow as it searches for ice. Following Image 1, render the dead acacia tree with the same forked branches, dry cracked bark, and hard midday sunlight; match that lighting on the bear so its white fur casts the same sharp shadows. Golden-orange dunes roll to the horizon under a clear blue sky."*

D.2 Evaluation protocol

Automated judge protocol. Each evaluation instance gives the judge the user prompt, the verification checklist, and the category-specific rubric, together with any reference images carrying short textual annotations (entity focus and severity). References are used only as comparison evidence; the single "image to assess" is scored. The judge follows a fixed sequence: recover task requirements; separate reference versus scored roles; extract observable evidence (subjects, attributes, scene, style, in-image text, artifacts); compare against references where checklist items depend on them; assign anchored half-point scores for checklist rows, rubric dimensions, and generic quality axes; then emit scored blocks for reference alignment, checklist, rubric, visual-reference fidelity, and text-reference fidelity, suitable for aggregation into the nine reported components (Table 8).

Score components and aggregation. All scores use a $[0, 3]$ scale in 0.5 steps (0 = not met/contradicted, 1 = weak with major issues, 2 = mostly met with minor issues, 3 = fully met). Each score is accompanied by an auditable reason in three parts: *Anchor* (initial score with evidence), *Adjustments* (signed deltas with justification), and *Computed final* (clamped sum matching the emitted score). Table 8 lists the nine reported components and their judge instructions. The overall score is the mean of present components, rescaled to 0–100.

Difficulty partition (Set I–III). The 651 search-intensive prompts are partitioned into three difficulty sets using Nano Banana Pro no-search quality as the sorting variable: each prompt receives an overall score from Nano Banana Pro *without any retrieval augmentation*, and tercile boundaries on this score distribution define Set I (easiest third), Set II, and Set III (hardest third). The partition is fixed across all experiments; the 100-prompt NoSearch subset is reported as a separate column.

Judge fidelity. The judge achieves Spearman $\rho = 0.87$ with human ratings on a held-out set of 500 prompt–image pairs (Appendix B.2). The Anchor/Adjustment/Computed diagnostic encourages falsifiable critiques of individual scores.

E Hardware and implementation details

All experiments were conducted on NVIDIA H20 (80 GB HBM3) GPUs. Below we detail the compute configuration for each training phase.

E.1 VLM reasoner training (Phase 0 and Phase 2)

The VLM reasoner (Qwen3-VL-8B) is trained using the ModelScope-Swift framework (`ms-swift`). We perform full-parameter supervised finetuning with the configuration in Table 9.

The SFT dataset consists of approximately 20,000 expert-annotated reasoning trajectories for the gate–filter–compress pipeline (Tasks A/B/C), stored in the training-row format consumed by ModelScope-Swift. A 1% held-out split is used for validation. We use dataset preprocessing with 200 workers and 4 dataloader workers for efficient data loading.

Table 8: **Judge scoring components.** Each component is scored on $[0, 3]$. Components 1–2 are prompt-specific; components 3–8 are fixed generic dimensions scored for every image; component 9 is conditional on whether reference images are provided.

#	Component	Judge instruction (abbreviated)
1	Checklist	Mean of per-item verification scores. Each prompt carries 3–10 binary-verifiable items (e.g., "Is the Om symbol correctly formed?"); the judge scores each against observable evidence.
2	Rubric adaptive	Weighted mean of prompt-specific rubric dimensions (e.g., *symbol_accuracy*, *historical_fidelity*) excluding the six fixed generic dimensions below.
3	Prompt faithfulness	Are all requested subjects, attributes, actions, scene, and style present and accurate?
4	Image quality	Visual clarity and sharpness; absence of artifacts or defects; style, lighting, and perspective coherence.
5	Text rendering	Does rendered in-image text match required content? Readable, correctly spelled, well-placed? Score left empty if no text is required and none is present.
6	AI naturalness	Organic texture realism vs. AI smoothness; fine-detail plausibility vs. uncanny uniformity; environment grounded vs. dreamlike or generic.
7	Composition & aesthetics	Framing and balance; depth and spatial arrangement; color harmony and consistent lighting; overall visual appeal.
8	Physical plausibility	Anatomy (fingers, limbs, proportions, pose); object physics (gravity, support, balance); spatial consistency (perspective, scale, occlusion); material properties; lighting and shadow consistency. Score left empty for abstract or stylized content.
9	Visual reference fidelity	Scored only when reference images are attached. Identity preservation (facial features, distinguishing traits); attribute consistency (colors, textures, proportions); style fidelity. Skipped (not zero) when no references are provided.

E.2 Generator DPO training (Phase 1)

Online iterative DPO for the visual generator (Flux.2-Klein-4B) is conducted using the Flow-Factory framework with the configuration in Table 10.

The DPO reward signal is provided by the Qwen3-VL-8B VLM judge, served via vLLM on a separate node (accessed at inference time over HTTP). The judge evaluates generated candidates using the SEARCHGEN-20K evaluation protocol with nine scoring dimensions. Judge inference uses a maximum of 4,096 output tokens, temperature 0.2, and a maximum pixel budget of 262,144 per image, with up to 24 concurrent requests.

Preference construction. For each prompt, the generator produces 5 candidate images (one additional candidate beyond the group for preference diversity via `preference_extra_candidates: 1`). Candidates are scored by the VLM judge, and preference pairs are constructed from the highest- and lowest-scored generations within each group. A structural similarity (SSIM) penalty (`dpo_penalize_condition_ssim: 0.95`) discourages degenerate pairs where chosen and rejected images are near-identical.

Flow-matching dynamics. We use a Flow-SDE scheduler with noise level 1.0, time shift 1.0, and timestep range truncated at 0.1 for stable training.

E.3 Reasoner rejection finetuning (Phase 2)

Phase 2 recalibrates the reasoner to the DPO-strengthened generator using rejection-sampling finetuning. The training infrastructure is identical to Phase 0 (ModelScope-Swift, $8\times$ H20, same hyperparameters), with the key difference that the training data consists of filtered trajectories: only reasoner rollouts that produce positive group-relative advantage (i.e., search improved the

Table 9: **VLM reasoner SFT hyperparameters** (Phase 0 and Phase 2).

Hyperparameter	Value
Base model	Qwen3-VL-8B-Instruct
Training type	Full-parameter SFT
Precision	bfloat16
Epochs	5–6
Learning rate	1×10^{-5}
Warmup ratio	0.05
Per-device batch size	1
Gradient accumulation steps	8
Effective batch size	64 (8 GPUs $\times 1 \times 8$ accum.)
Max sequence length	12,288 tokens
Max image pixels	262,144 (512^2)
Image max token number	512
Attention implementation	FlashAttention-2
Sequence packing	Enabled (padding-free)
Gradient checkpointing	Enabled
ViT parameters	Frozen
DeepSpeed stage	ZeRO-2
Hardware	$8\times$ NVIDIA H20 (80 GB)
Training data	$\sim$20K SFT trajectories
Wall-clock time	$<$4 hours

strengthened generator's output quality) are retained for training. The full RFT cycle completes in approximately $4 \times 8 = 32$ GPU-hours.

E.4 Total compute budget

Table 11 summarizes end-to-end training compute (Phase 1 GPU-hours are 8 GPUs $\times$24 h wall-clock). The VLM judge service runs concurrently on a separate GPU and is not folded into the training GPU-hour totals.

All training runs use CUDA memory optimization via `PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True`. Distributed training communication uses NCCL over NVLink interconnects within each 8-GPU node.

F Limitations and future directions

Our work deliberately prioritizes depth of insight over exhaustive optimization, and this choice opens several promising avenues for future research.

Iterative co-training. Our recipe consists of a single DPO pass followed by a single RFT pass. While this minimal protocol already produces monotonic improvement and validates the core principle, the knowledge boundary between internalizable and contextual knowledge is unlikely to be fully resolved in one iteration. Multi-round co-training—where the generator and reasoner alternate improvement over several cycles—could progressively sharpen this boundary and unlock additional gains. Investigating convergence behavior, diminishing returns, and potential instability in such iterative loops is an exciting direction.

Generator scale and architecture. Our co-training experiments use a 4B-parameter generator (Flux.2-Klein-4B) and 7B-parameter generator (Bagel). The substantial gap between our best result and frontier commercial systems suggests that larger or architecturally different generators may internalize a broader swath of knowledge, shifting the knowledge boundary and potentially changing the distribution of prompts that require search. Studying how the internalizable/contextual split evolves across generator scales—and whether certain failure categories (e.g., temporal-recent vs. cultural-specificity) exhibit qualitatively different scaling behavior—would deepen understanding of the knowledge boundary as a function of model capacity.

Table 10: **Generator DPO hyperparameters** (Phase 1).

Hyperparameter	Value
Base model	Flux.2-Klein-4B
Finetuning type	LoRA
LoRA rank / alpha	64 / 128
Target modules	Default (attention layers)
Precision	bfloat16
Learning rate	1×10^{-4}
Optimizer	AdamW (β_1=0.9, β_2=0.999, ε=10^{-8})
Weight decay	1×10^{-4}
Max gradient norm	1.0
Resolution	512×512
Condition image size	512×512
Per-device batch size	1
Group size (candidates per prompt)	5
Unique samples per epoch	64
Gradient steps per epoch	2
Gradient accumulation	Auto (computed from group size)
DPO β	100.0
Reference model device	CPU (offloaded)
Advantage aggregation	Group-relative DPO
EMA decay / update interval	0.99 / 4 steps
Guidance scale	4.0
Inference steps (sampling)	50
Timestep sampling	Logit-normal (μ=0.0, σ=1.0)
Gradient checkpointing	Enabled
Hardware	8× NVIDIA H20 (80 GB)
Training data	∼7K prompt–image preference pairs
Save frequency	Every 10 gradient steps

Table 11: **Compute budget overview.** Training GPU-hours are approximate.

Phase	GPUs	Wall-clock	GPU-hours
Phase 0: Reasoner SFT	8× H20	<4 h	∼32
Phase 1: Generator DPO	8× H20	24 h	∼192
Phase 2: Reasoner RFT	8× H20	∼4 h	∼32
VLM Judge serving (vLLM)	1× H20	concurrent	—
Total (training)	—	—	∼256

Reward signals. Our current evaluation relies on an automated VLM judge (Gemini-3-Flash). While it achieves strong correlation with human ratings (Spearman $\rho = 0.87$), such judges can introduce noisy rewards—a widely acknowledged issue when automated scoring substitutes for human judgment [45, 25]. The choice of DPO is intentional: it resists reward noise by selecting the best- and worst-ranked generations to construct preference pairs [32].

G Broader impact

This work contributes positively to the research community and society in several important ways, while also carrying risks that warrant consideration.

Positive impacts.

- **New evaluation resources for an underserved problem.** SEARCHGEN-20K and SEARCHGEN-BENCH fill a significant evaluation gap. Existing text-to-image benchmarks predominantly test prompts that fall within generators' parametric knowledge, creating an illusion of near-solved performance. By systematically cataloging twelve failure categories grounded in over 10,000 real user prompts, our benchmark surfaces a class of failures

that affects daily production use but has been largely invisible to the research community. Making this gap measurable is a prerequisite for closing it.

- **A long-tailed, culturally diverse, and challenging reference-to-image dataset.** The released corpus of 20,000 prompts, spanning twenty-two domains and intentionally representing global cultural diversity—including underrepresented cultural symbols, regional artifacts, non-Latin typography, and entities from the Global South—provides a uniquely challenging testbed. The long-tailed entity distribution mirrors real user demand and offers the community a resource for studying knowledge-intensive generation at a scale and diversity not previously available. The accompanying pre-executed search results (both image and web returns) form a fully offline search environment, lowering the barrier for reproducible research on search-augmented visual generation without requiring live API access or incurring search costs.
- **Advancing equitable visual generation.** A disproportionate share of search-intensive failures affects culturally specific content: regional festivals, indigenous art forms, historical figures outside Western canons, and scripts beyond Latin. By explicitly benchmarking these categories and demonstrating that search-augmented generation can improve fidelity for such content, this work contributes toward more equitable visual AI that can serve diverse global users rather than only those whose cultural knowledge is well-represented in training data.
- **Computational accessibility.** The deliberately minimal co-training recipe (one DPO pass, one RFT pass, a 4B generator, an 8B reasoner, fitting in 4×8 GPU hours for the RFT phase) demonstrates that meaningful progress on knowledge-intensive generation does not require frontier-scale compute, broadening access to researchers with limited resources.

Potential negative impacts and mitigations.

- **Misinformation risk.** Improving generators' ability to render specific real-world entities—public figures, cultural symbols, historical events—could lower the barrier for producing convincing visual misinformation. We note, however, that our benchmark explicitly includes an "AIGC fakeness" evaluation dimension that measures detectable artifacts, and our released assets are evaluation prompts and search results rather than trained generator checkpoints. We encourage downstream users to pair improved generation with provenance tracking and watermarking.
- **Privacy concerns.** Some search-intensive prompts involve public figures. Our dataset construction follows standard practices: all seed entities are drawn from publicly available information, and the benchmark is designed for research evaluation rather than generating deceptive impersonations. We will include usage guidelines with the released dataset that prohibit malicious identity synthesis.
- **Bias amplification.** While we intentionally design for cultural diversity, the search results that form part of our released offline environment inherit biases from web search engines. Researchers using these results should be aware that search-returned references may reflect stereotypical or incomplete representations of certain cultures, and should evaluate whether such biases propagate into generated outputs.

G.1 Third-party models, data access, and terms of use

Open-weight generators. Bagel [12]; FLUX.2 Klein checkpoints [4]; Qwen-Image [46]. **Commercial APIs.** Qwen-Image-2 [46]; Gemini developer endpoints (Gemini 2.5 Flash Image, Gemini 3 Pro Image; consumer names Nano Banana / Nano Banana Pro in some regions) [15]; ByteDance Jimeng [7]; SeedDream 4.0 [8]; Imagen 3 Fast [3]. Public documentation does not specify whether these endpoints internally invoke retrieval, tool use, or proprietary knowledge bases at inference time. Consequently, high SEARCHGEN-BENCH scores for commercial systems should be interpreted as end-to-end product behavior rather than as pure comparisons to open-weight generators under identical conditioning assumptions.

Use. Experiments respect each provider's publicly posted terms and documentation; releases from this work consist of prompts and cached search artifacts, not redistributed model weights or proprietary API payloads.

G.2 Responsible release and safeguards

Release scope. SEARCHGEN-20K and SEARCHGEN-BENCH are intended for research evaluation of knowledge-intensive visual generation. The public release will include prompts, annotations, evaluation checklists, rubrics, search queries, and permissible search metadata or derived attributes, rather than raw scraped images or proprietary web content. We do not redistribute copyrighted source images, proprietary web content, or proprietary API payloads without permission.

Safeguards and intended use. To reduce copyright and safety risks, we filter or exclude unsafe search artifacts, including explicit sexual content, graphic violence, hateful or extremist material, and personally sensitive content. The benchmark is intended for evaluating search-intensive visual generation, not for producing deceptive, infringing, or harmful content. Users of the released assets should respect the licenses and terms of the original sources and should not use the benchmark to generate unauthorized reproductions of protected characters, brands, or individuals.