BENCHMARKING AND EVOLVING THE CAPABILITY BOUNDARY IN AGENTIC VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual generators render complex scenes convincingly, yet they confidently fabricate the characters, symbols, and facts they never learned. This *world-knowledge bottleneck* is structural: generators learn from fixed corpora, whereas user requests are unbounded, evolving, and long-tailed. To measure and study it, we construct the first large-scale dataset **SEARCHGEN-20K, SEARCHGEN-CORPUS-1M** and **SEARCHGEN-BENCH** featuring 20,937 production-level generation tasks paired with truth-grounded verifiers that check the requested content rather than image quality alone. Comparing open-weight generators with agent-powered frontier APIs, our analysis reveals a substantial gap of up to 35pp in handling production-level requests. Agentic visual generation, in which an agent retrieves evidence for the generator, is the natural remedy. Yet we identify two challenges: naive search can override generator capabilities due to search noise, and agent become stale when generator evolves its capability. In response, we showcase a minimal baseline approach that introduces a noise-resistant agentic protocol and co-adapts the generator and the agent. **Data, code, and models are released for future research**.

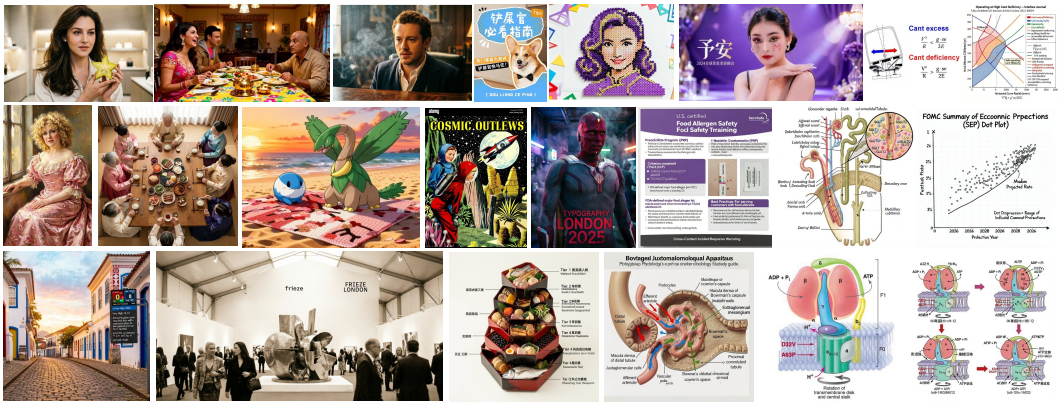


Figure 1: **Constructed requests and search-augmented generations from SEARCHGEN-20K.** Examples from twelve categories of 20,840 production prompts show visual, textual, and mixed evidence needs.

1 INTRODUCTION

Ask a frontier image generator for the mascot of the 2025 Osaka Expo, and you may get a polished, confident fabrication. Ask for a historically accurate Spartan phalanx, and you may get anachronistic armor rendered in exquisite detail. These generators already render complex scenes with precise lighting and coherent structure (Black Forest Labs et al., 2025; Esser et al., 2024); what they lack is knowledge of specific identities, appearances, and facts (Figure 1). This *world-knowledge bottleneck* is structural: generators learn from fixed corpora, whereas user requests are *unbounded, evolving, and deeply long-tailed*. Like an illustrator facing an unfamiliar subject, a generator needs external inputs, e.g., web knowledge, visual references, not better rendering alone.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Measuring the bottleneck requires telling an attractive image from a correct one. From 20,840 production prompts, we construct SEARCHGEN-20K, 20,937 tasks whose prompt-specific checks test the requested character, symbol, or historical detail, and hold out SEARCHGEN-BENCH for evaluation. Under these checks, generators that look alike on familiar requests come apart: Qwen-Image and Nano Banana Pro nearly tie on NoSearch prompts (70.0 versus 73.3) but differ by 35 points on Search-Intensive prompts, where all eight open-weight generators score 17.5–31.6 (Figure 3). Looking convincing is not the same as getting the requested content right.

The obvious remedy is *agentic visual generation systems*, e.g., GPT-Image-2 (OpenAI, 2026), in which an agent retrieves external evidence and prepares it for the generator. Yet blind search, which retrieves for every nominated knowledge gap, can hurt: it raises Search-Intensive scores but lowers Qwen-Image-2 from 70.7 to 60.4 on NoSearch prompts (Table 3). The cause is *search noise*: a relevant reference also carries poses, backgrounds, or layouts the user never requested, and the generator copies them along with the missing detail (Figure 4). Retrieval relevance does not guarantee generating the right image.

Search noise makes the value of external inputs depend on the generator that receives it: a visual reference or web knowledge may contrarily distract a model that already knows it. An agent must therefore decide, for a given generator, whether to search, which results to keep, and which attributes to borrow, and revisit these decisions as training changes the generator. Learning them requires observing their consequences, so we collect and release SEARCHGEN-CORPUS-1M, 159,027 cached search sessions that every agent replays identically, and link 96,848 reasoning traces in SEARCHGEN-20K to the 283,493 images they produced (Table 2). Together, they establish a interactive environment for training generation agents. All assets are released to enable reproducible training and further research.

Building on this environment, we show that a deliberately minimal baseline already improves agentic visual generation. Our approach first introduce a *noise-resistant agentic protocol*, which gates search on gaps the agent rates critical or important, filters for references that supply the missing detail with little else, and integrates each by naming the attribute to borrow (Section 3.1). We then *co-train* the two components: online iterative DPO updates the generator on search-augmented inputs, and rejection-sampling finetuning adapts the agent to the updated generator (Section 3.2). On Klein-4B, Search-Intensive quality continues to improve throughout the co-training, reaching a 20% relative improvement; Bagel follows the same pattern (Table 5).

Contributions. Our analysis confirms the world-knowledge bottleneck and pinpoints agentic visual generation as a promising research direction. We therefore release SEARCHGEN-20K and SEARCHGEN-BENCH, the first large-scale dataset featuring production-level tasks with prompt-specific verifiers, and SEARCHGEN-CORPUS-1M, backing the environments for generation agent training; all are released. We also reveal the key challenges of how search noise can harm generation, facilitating deeper understanding of the new research problem. To set a baseline for the problem, we propose a minimalist co-training approach, which already demonstrates evolving capability in agentic visual generation across different generators.

2 THE WORLD-KNOWLEDGE BOTTLENECK

An image may be visually convincing yet depict the wrong character, symbol, or historical detail. Search can supply what the generator lacks, but retrieved content can also distort an image that would otherwise satisfy the request. To study both possibilities, we release SEARCHGEN-20K with tasks, prompt-specific verification criteria, reasoning traces, and generated images; SEARCHGEN-CORPUS-1M with indexed, cached image and web search results; and SEARCHGEN-BENCH as a held-out evaluation of those tasks. Together, the tasks and verifier define success, while the replayable corpus supplies an offline environment in which generation agents can learn to search and use evidence. This section explains how these are created and what value they provide.

2.1 FROM USER REQUESTS TO VERIFIABLE TASKS

We analyze 20,840 production text-to-image prompts and identify twelve recurring categories of knowledge-dependent requests (Table 1; Figure 1). What matters across these categories is the evidence a request needs: a character's appearance calls for visual reference, historical dates call

Table 1: **Taxonomy of search-intensive visual generation.** Twelve categories by failure source, each with an example prompt. *Modality* gives the dominant knowledge type.

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	"Traditional Oaxacan alebrije dragon with authentic colors"
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	"Nostalgia on a rainy afternoon in a small Japanese town"
Implicit Reasoning	Both	"Cozy mountain cabin interior in the style of a Miyazaki film"

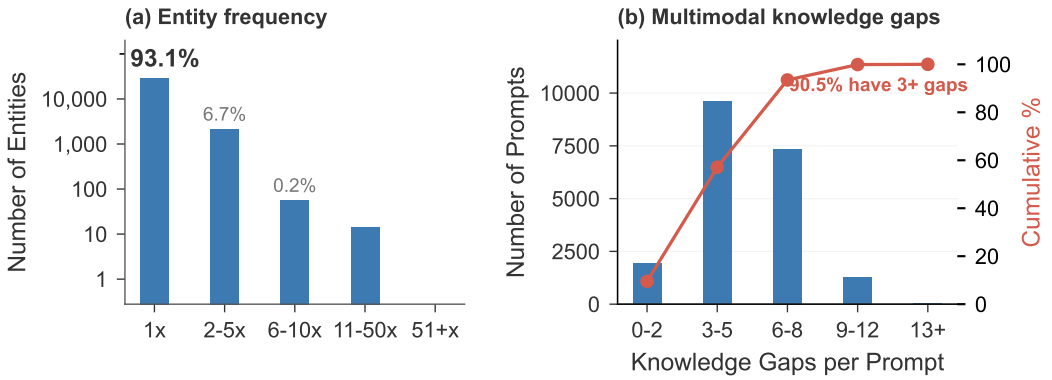


Figure 2: **Entity rarity and multi-gap prompts in SEARCHGEN-20K.** (a) Of 31,537 entities, 93.1% occur in one prompt. (b) Prompts have 5.2 knowledge-related annotations on average (visual slots, text slots, and failure modes); 90.5% have at least three.

for textual facts, and a recent mascot may call for both. This distinction guides the knowledge annotations and the choice of search modality.

SEARCHGEN-20K turns production request patterns into verifiable tasks. We decompose 20,840 production prompts into entity seeds, request patterns, compound constraints, and visual or textual knowledge requirements. From these ingredients, we employ answer-first synthesis and acquire 20,937 production-level tasks across twenty-two domains in English and Chinese. These tasks are paired with known truths of knowledge gaps by construction. This construction also preserves the identified task types and combinations of constraints while prioritizing coverage of long-tail cases. Of these tasks, 20,188 are used for training and validation. The 749-prompt SEARCHGEN-BENCH test set is disjoint from the generator and agent training prompts. Every task in the dataset and benchmark records the knowledge each task requires, its target contents, and prompt-specific checklists and rubrics for the generated image.

The tasks also retain the compound and long-tailed character of user demand. Prompts carry 5.2 knowledge-related annotations on average, including failure-mode labels, while 93.1% of 31,537 entities appear in only one prompt *within* SEARCHGEN-20K (Figure 2). These counts describe annotation density and dataset frequency; they do not measure independent knowledge gaps or what a generator saw during training. For the experiments here, the annotations specify what evidence may be needed and what the image must get right.

2.2 AN OFFLINE ENVIRONMENT FOR TRAINING GENERATION AGENTS

Tasks and verifiers define what an agent should accomplish, but training it to search also requires a retrievable environment. **SEARCHGEN-CORPUS-1M provides environments for agentic training:** we release the complete indexed archive of 159,027 pre-executed image and web search ses-

Table 2: **Released Assets.** For each training prompt, agentic reasoning traces collect from multiple frontier reasoning model APIs and generator APIs, paired with verifier scores. Million-scale search logs are released to establish interactive environment for agent training.

Asset	Scale	Enables
Training/validation trajectories *(20,188 unique prompts)*		
Reasoning traces (frontier models)	96,848	search-policy / reward learning
Generated images (trace–model pairs)	283,493	preference & distillation data
Search corpus *(archived, indexed)*		
Search sessions	159,027	reproducible retrieval, no live API
image / web	120,828 / 38,199	
SERP hits (unique URLs)	1,097,961	retrieval / grounding studies
Cached downloads	705,838	replay where content is cached
image / web	599,859 / 105,979	

sions, with over one million unique result URLs and 705,838 cached downloads (Table 2). Search-session indices and replay code let an agent retrieve recorded results for archived queries without a live search API. Cached content lets researchers rerun reference selection and integration with fixed search results across training runs. All SEARCHGEN-BENCH results in Section 4 use this replay, so every compared agent sees the same evidence.

SEARCHGEN-20K also supplies the decisions and outcomes needed to learn from that environment. We release 96,848 reasoning traces and 283,493 generated images linked to its tasks and archived searches (Table 2). Traces can supervise when and how to search; images show the consequences of those decisions and support preference learning or distillation. Together with the verifier, these linked assets connect a request, a search action, retrieved evidence, and the resulting image in one training and evaluation loop. Offline replay covers the indexed archived queries and cached content, rather than arbitrary new web searches.

The verifier must score the requested detail, not just general image quality. SEARCHGEN-BENCH applies prompt-specific checklists and rubrics alongside general image-quality measures (Appendix D.3). For example, a depiction of Yang Chaoyue in 1970s rural China is checked for recognizable identity, period accuracy, and the requested tool and style; pleasing film grain cannot substitute for the wrong person or setting. To enable stable evaluation, we finetune a 8B VLM judge. Gemini-3-Flash first provided scoring supervision for approximately 400,000 trajectories drawn from the training prompt pools. We supervised-finetuned a Qwen3-VL-8B VLM judge on these structured reasoning and scoring responses (Wang et al., 2026b). This judge scores the images used for all reported results with greedy decoding and reports nine components (Table 8): two prompt-specific measures, six generic measures, and visual-reference fidelity when applicable. The unweighted mean of applicable 0–3 scores is rescaled to 0–100. We refer the reader for the evaluation details to Appendix D.3 and the released code.

2.3 WHAT THE BENCHMARK REVEALS

Our retrieval and co-training experiments use two selected diagnostic strata: 100 NoSearch prompts for which unaided generation was judged adequate and 649 Search-Intensive prompts selected for external knowledge needs, for 749 prompts in total. Because adequacy helps define NoSearch, this contrast is descriptive; the strata do not estimate their prevalence in production traffic. Figure 3 presents the expanded 18-generator comparison from the public benchmark release; Tables 4 and 6 report the component scores from our 749-prompt experimental subset.

In the expanded comparison, eight open-weight generators score 17.5–31.6 on Search-Intensive prompts, well below their 34.4–70.0 range on NoSearch (Figure 3). Qwen-Image and Nano Banana Pro score 70.0 and 73.3 on NoSearch, respectively, but 28.6 and 63.6 on Search-Intensive prompts, a 35.0-point difference in the latter stratum. **Familiar-request performance therefore masks a substantial difference on knowledge-intensive requests.** Component scores from our 749-prompt experimental subset show why the distinction matters: for Flux.2-Klein-9B, Image Quality is 36.8, but Checklist and Visual Reference scores are 24.2 and 16.9 (Table 4). Its Text Rendering score of 7.2 also shows that some errors involve rendering. These cross-stratum results locate the difficulty; they do not, by themselves, show whether search resolves it.

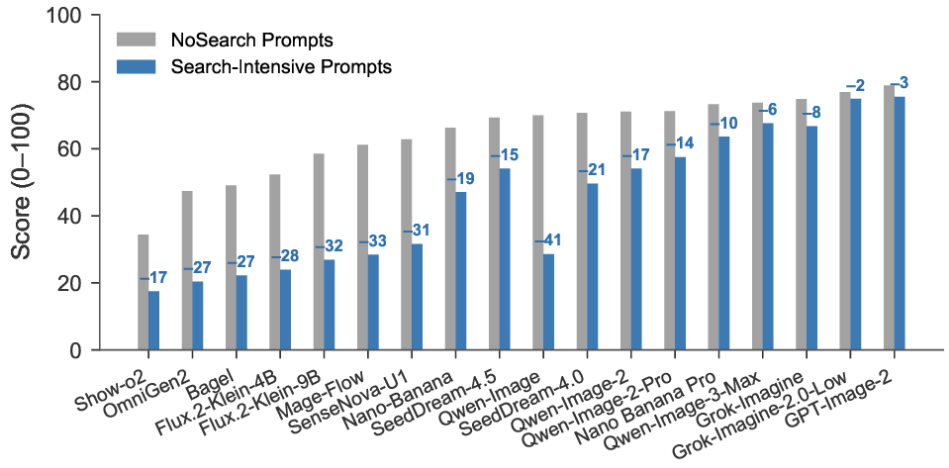


Figure 3: **Overall quality by knowledge stratum across 18 generators in the public benchmark release.** Gray and blue bars show the nine-component mean on NoSearch and Search-Intensive prompts, respectively. Open-weight generators show large declines; Agent-powered frontier APIs such as GPT-Image-2 expose only tiny gaps on search-intensive promtps.

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

Table 3: **Search is a double-edged sword.** PostHoc selects the higher-scoring output after generating both searched and unsearched candidates.

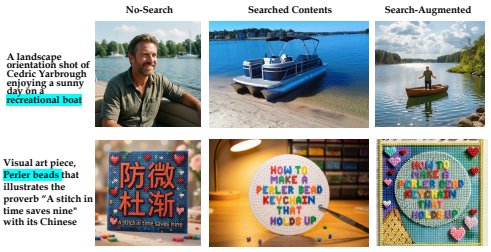


Figure 4: **Failure examples of search.** Search queries executed are colored. Searched contents introduce noise and may degrade generation quality.

We next compare searched and unsearched generation on the *same prompts*. BlindSearch searches for nominated gaps without first deciding whether retrieval is useful to the generator. The 649 Search-Intensive prompts are divided by evidence need into VisualSearch (386) and TextualSearch (263), while NoSearch is evaluated separately. PostHocSelect provides a further diagnostic reference by generating both candidates and choosing afterward, at the cost of two generations and scoring.

BlindSearch improves both search-intensive strata for all three tested generators, but lowers NoSearch quality (Table 3). For Qwen-Image-2, VisualSearch rises from 37.2 to 45.3 while NoSearch falls from 70.7 to 60.4. **The same retrieval policy can help or harm across the selected request groups.** Harm is not confined to NoSearch: PostHocSelect exceeds BlindSearch in every stratum for all three generators, so on some search-intensive prompts the unsearched image is still better. Figure 4 shows how this *search noise* arises: a boat reference can redirect the subject and composition, while a bead-art reference can import unwanted lettering and layout. The issue is thus both whether to retrieve and which parts of retrieved content to pass on.

These findings turn knowledge-intensive generation into three measurable decisions: when external evidence is needed for a *particular generator*, which results contain the useful detail, and how to use that detail without importing unrelated content. The first decision may change when the generator is trained, but the diagnostics here do not measure such a change. Section 3 addresses these decisions with a noise-resistant agentic protocol and co-trains the agent with the generator.

3 Co-Training Agentic Visual Generation

Analysis in Section 2 indicates three decisions for agentic generation systems: whether a particular generator needs external inputs, which results carry the missing detail, and how to pass that detail

Table 4: **Component scores on the 649 Search-Intensive prompts in our experimental subset.** All scores are on a 0–100 scale. Checklist, Rubric, and Visual Reference assess requested details; Text Rendering, Physical Plausibility, and Image Quality show other aspects of output quality. Open-weight systems score especially low on the requested-detail components.

	Knowledge-Sensitive			*Other Quality Dimensions*		
Generator	**Checklist**	**Rubric**	**Visual Reference**	**Text**	**Physical**	**Img Quality**
Open-weight generators						
Bagel (Deng et al., 2025)	18.2	17.6	13.5	2.5	36.8	30.5
Flux.2-Klein-4B (Black Forest Labs, 2026)	19.8	18.4	11.9	4.2	46.2	37.2
Flux.2-Klein-9B (Black Forest Labs, 2026)	24.2	23.1	16.9	7.2	48.6	36.8
Qwen-Image (Wu et al., 2025)	24.8	24.3	17.7	8.7	44.6	40.1
Commercial systems						
Qwen-Image-2 (Zhao et al., 2026)	28.5	27.1	21.0	12.7	48.2	42.2
SeedDream-4.0 (ByteDance Seed Team, 2025)	44.2	43.6	35.1	35.9	64.0	57.0
Nano Banana (Google DeepMind, 2025)	41.0	40.4	33.2	28.0	65.5	57.1
Nano Banana Pro (Google DeepMind, 2025)	64.4	63.1	58.3	65.0	78.5	71.4
GPT-Image-2 (OpenAI, 2026)	71.2	70.1	66.0	75.9	77.3	75.1

on without the rest. Our noise-resistant agentic protocol makes each decision an explicit step (Section 3.1). The first decision, however, depends on the generator's *capability boundary*: the request distribution that can be handled adequately without external inputs (Appendix D.4). Training the generator can move this boundary, so an agent fitted to the old generator can become stale. We therefore co-train the two: train the generator on the agent's evidence, then adapt the agent to the updated generator (Section 3.2; Figure 5). Each step uses a standard objective (SFT, DPO, or rejection finetuning), which keeps the baseline minimal; what it adds is the protocol and the order in which the two components learn from image outcomes.

3.1 A Noise-Resistant Agentic Protocol

Search noise can enter at each decision, through an unnecessary search, a distracting result, or an overused reference. The agent therefore takes three steps in order (Figure 5; per-step inputs, outputs, and we provide worked examples in Appendix D.2).

Gate decides whether to search, from the prompt alone and before any retrieval. It comphrehends raw user requests, lists knowledge gaps with a severity level and a tentative search query for each. Only knowledge gaps rated `critical` or `important` will issue search with its binding search query: image queries for appearances and visual shortcuts, web queries for facts and knowledge. Unlike BlindSearch, which retrieves for every nominated gap (Table 3), the gate issues no query when no gap qualifies the threshold.

Filter decides which result to keep. Given retrieved images from a image search query, it returns the single reference that supplies the missing detail with the least unrelated content: among photographs of the right costume, it prefers one whose pose and background will not compete with the user's scene. Web queries contribute the text snippet each result provides as a summary of its page.

Integrate decides what information to borrow from these external sources. From the prompt, the selected references, and the web snippets, it states what to borrow from each reference and what to create anew, then writes a refined prompt that keeps the user's scene, states the retrieved facts, and names each borrowed attribute, such as a teal-and-gold robe. The generator receives this prompt with the visual references, so the instruction points it to one attribute within a richer image.

3.2 Co-Training: Train the Generator, Then Adapt the Agent

A protocol trajectory, from gate decision to specification, can help one generator and mislead another; its value shows only in the image it produces. Every training signal below therefore comes from generated images scored against the original request. Algorithm 1 illustrates the complete procedure.

Phase 0: supervised warm-start. We finetune Qwen3-VL-8B (Bai et al., 2025) on 20,080 Gemini-3-Flash traces from SearchGen-20K, keeping the highest-scoring trajectory per training prompt and

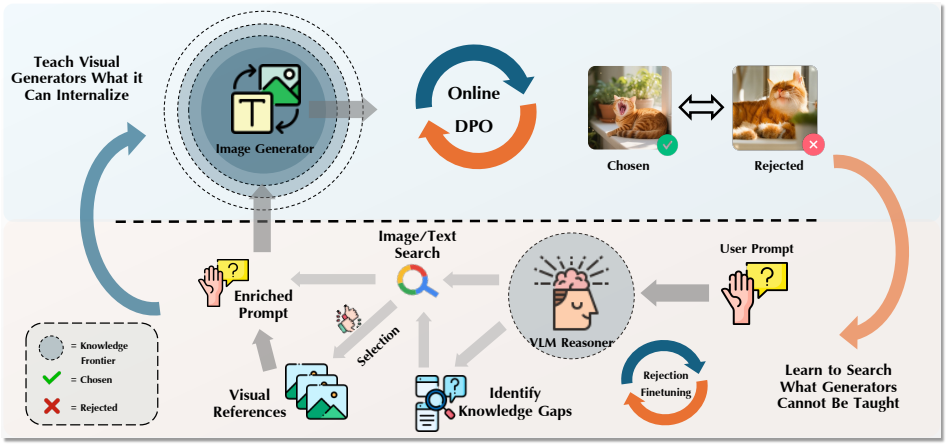


Figure 5: **Co-training for agentic visual generation.** The VLM agent gates search on knowledge gaps, filters image or web results, and integrates the selected evidence into the generator input. Phase 0 warm-starts the agent by SFT; Phase 1 trains the generator by DPO on preference pairs from search-augmented generations; Phase 2 adapts the agent to the Phase 1 generator by RFT on trajectories that score above their sampled group.

splitting it into gate, filter, and integrate steps (Appendix E.2). The resulting SFT AGENT follows the protocol but is not adapted to either generator we train.

Phase 1: generator preference training. The SFT AGENT builds search-augmented inputs for each training prompt; the generator samples several images per input, the same finetuned Qwen3-VL-8B judge scores them against the original request, and the highest- and lowest-scored images form a preference pair. Because a candidate that copies a reference's pose or layout tends to fail the request's checks, online iterative DPO (Rafailov et al., 2023; Wallace et al., 2024) on these pairs favors using evidence over copying it (Klein-4B objective in Appendix E.3). The update can also move the generator's capability boundary; Section 4 measures the change in unaided generation.

Phase 2: agent adaptation. The updated generator may now render some requests unaided and respond differently to the same evidence, so the SFT AGENT's decisions can become stale. We sample several trajectories per prompt from the SFT AGENT, execute each with the Phase 1 generator, and keep those scoring above their group mean; rejection-sampling finetuning (RFT) on them yields the ADAPTED AGENT. Each kept trajectory fixes a gate decision, references, and an integration instruction, so RFT adjusts their joint effect on the image; it does not directly compare searched with unsearched outputs.

Conceptually, the co-training cycle can be iterated to evolve the joint agentic generation system. In practice, we run one pass of each phase as the minimalist baseline. Section 4 evaluates each update with the other component fixed and pairs the ADAPTED AGENT with the base generator to test whether its gain depends on the generator that supplied its feedback.

4 EVOLVING THE CAPABILITY THROUGH CO-TRAINING

Section 2 raised two challenges: search noise can harm requests a generator already handles, and an agent fitted to one generator can become stale once that generator is trained. We therefore ask whether each co-training stage adds gains and whether the protocol avoids the cost of BlindSearch (Section 4.1), and whether the ADAPTED AGENT's gain depends on the updated generator (Section 4.2).

Setup. We co-train two generators with different architectures, Flux.2-Klein-4B (**Klein-4B**), a flow-matching model, and Bagel-7B (**Bagel**), a unified vision–language model, so that a shared pattern is not an artifact of one architecture. Both accept text and reference images. Following Section 3.2, each phase changes one component: Phase 0 pairs the SFT AGENT with the base generator, Phase 1 swaps in the DPO generator, and Phase 2 swaps in the ADAPTED AGENT. Reference rows pair the DPO generator with generator-only inference or with Gemini-3-Flash as a frontier agent under the same protocol, and pair the ADAPTED AGENT with the base generator. All agents replay

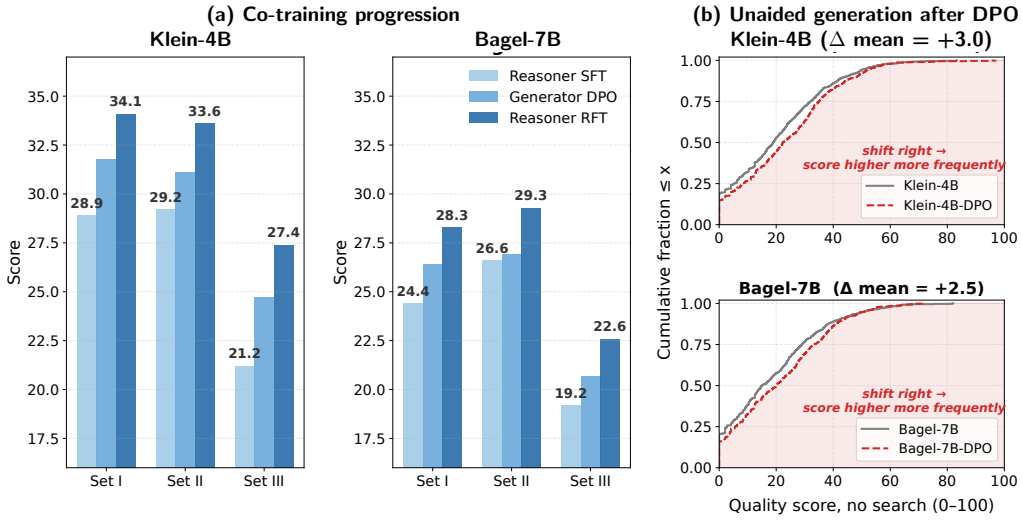


Figure 6: **Each co-training stage adds gains, and DPO shifts unaided generation.** (a) Search-Intensive quality rises from the SFT AGENT through generator DPO to the ADAPTED AGENT in every difficulty tier for both generators. (b) Cumulative distributions of unaided scores before and after DPO for Klein-4B (top) and Bagel (bottom); rightward shifts indicate higher unaided scores after generator training.

SEARCHGEN-CORPUS-1M (Section 2.2), so they see identical evidence. SEARCHGEN-BENCH comprises 649 Search-Intensive and 100 NoSearch prompts, scored 0–100 by the verifier of Section 2.2. To test whether gains reach the hardest requests, we split Search-Intensive prompts into fixed terciles by Nano Banana Pro's unaided score, from Set I (easiest) to Set III (hardest).

4.1 EACH CO-TRAINING STAGE ADDS GAINS

Both stages raise Search-Intensive quality for both generators (Table 5). On Klein-4B, generator DPO lifts the score from 26.4 to 29.2 and agent adaptation to 31.8, a 20% relative gain; Bagel rises from 23.4 to 24.7 to 26.8. Each step holds the other component fixed, so the Phase 2 gain cannot come from further generator training. Every difficulty tier improves at each stage, including Set III (Figure 6a), so the gains are not confined to requests the generator nearly handles already.

The protocol also avoids the cost BlindSearch paid on NoSearch prompts. BlindSearch lowered NoSearch quality by 5.5–10.3 points for all three generators in Table 3; here the Klein-4B-DPO system scores 54.0 with the SFT AGENT and 56.9 with the ADAPTED AGENT, against 49.9 for generator-only inference, and Bagel shows the same ordering. On Search-Intensive prompts, the final system gains 6.8 points over generator-only inference. Because Table 3 uses different generators and both comparisons use aggregate scores, they cannot separate skipped searches from better-used evidence or prompt refinement. Search noise is also not eliminated: with Klein-4B-DPO, both agents still trail generator-only inference on TextualSearch prompts (Appendix B.1).

4.2 ADAPTING THE AGENT TO THE EVOLVED GENERATOR

Generator DPO changes the capability boundary that the SFT AGENT was fitted to: unaided score distributions shift rightward for both generators (Figure 6b). After DPO, NoSearch quality with the SFT AGENT dips slightly (54.6 to 54.0 on Klein-4B), whereas the ADAPTED AGENT raises it to 56.9; Bagel follows the same pattern.

If the ADAPTED AGENT were simply a better agent, its gain would transfer to the base generator. It largely does not: relative to the SFT AGENT, it adds 2.6 points with Klein-4B-DPO but only 0.4 with base Klein-4B, and 2.1 versus 0.7 on Bagel (Appendix C.2). Most of the adaptation gain is therefore specific to the generator that supplied its feedback. This is the staleness co-training addresses: once the generator is trained, its agent must be adapted again.

Under the same generator and protocol, the adapted 8B agent scores 31.8 versus 31.2 with Klein-4B-DPO and 26.8 versus 26.1 with Bagel-DPO when compared with the Gemini-3-Flash agent, which

Table 5: **Each co-training stage raises Search-Intensive quality for both generators.** Phase 0 pairs the SFT AGENT with the base generator; Phase 1 trains the generator with DPO; Phase 2 adapts the agent with RFT. Reference rows use generator-only inference, a Gemini-3-Flash agent, or the ADAPTED AGENT with the base generator. Search-Intensive prompts are split into fixed difficulty terciles (Set I: easiest; Set III: hardest).

		NoSearch	Search-Intensive			
			Set I	Set II	Set III	Overall
Co-training progression						
Klein-4B						
Phase 0	SFT AGENT + Klein-4B	54.6	28.9	29.2	21.2	26.4
Phase 1	SFT AGENT + Klein-4B-DPO	54.0	31.8	31.1	24.7	29.2
Phase 2	ADAPTED AGENT + Klein-4B-DPO	**56.9**	**34.1**	**33.6**	**27.4**	**31.8**
Bagel						
Phase 0	SFT AGENT + Bagel	52.6	24.4	26.6	19.2	23.4
Phase 1	SFT AGENT + Bagel-DPO	52.4	26.4	26.9	20.7	24.7
Phase 2	ADAPTED AGENT + Bagel-DPO	**54.3**	**28.3**	**29.3**	**22.6**	**26.8**
Reference baselines						
Klein-4B						
	Generator only + Klein-4B-DPO	49.9	28.2	26.3	20.6	25.0
	GEMINI-3-FLASH + Klein-4B-DPO	55.7	33.7	33.9	26.0	31.2
	ADAPTED AGENT + Klein-4B	54.6	29.0	29.8	21.5	26.8
Bagel						
	Generator only + Bagel-DPO	48.8	24.3	24.0	18.8	22.4
	GEMINI-3-FLASH + Bagel-DPO	53.2	28.1	29.0	21.1	26.1
	ADAPTED AGENT + Bagel	52.5	25.1	27.3	19.8	24.1

is not adapted to either generator. These results close only a small part of the gap in Section 2.3: GPT-Image-2 scores 71.0 on the same Search-Intensive prompts (Appendix C.1). Aggregate scores also do not show which protocol decision changed; Appendix D.5 tests whether DPO changes which requests benefit from fixed evidence and whether the adapted gate tracks that change.

5 RELATED WORK

Appendix G provides a full discussion of related work. Retrieval-augmented image generators condition on external image or image–text evidence (Chen et al., 2023; Blattmann et al., 2022; Ashual & Wolf, 2023). Closest to our setting, Gen-Searcher trains a multimodal search agent while keeping the image generator fixed (Feng et al., 2026). We focus on benchmarking agentic visual generation systems on production-level tasks, and propose co-adapt agent and generator (§3.2).

Selective retrieval in language models motivates deciding when to consult external evidence (Asai et al., 2024; Jiang et al., 2023). Our analysis (Table 3) reveal that naive search can hurt generation because they inject noise that override generator's capabilities. Compositional and faithfulness benchmarks (Li et al., 2024; Huang et al., 2023; Lee et al., 2023) address related evaluation needs; SEARCHGEN-BENCH focus on real-world production-level tasks, and pair each task with prompt-specific truth-grounded verifiers.

6 CONCLUSION

SearchGen releases the first production-level large-scale dataset and benchmark, paired with a search corpus that establishes environments for agentic training. Our analysis revealed open-weight generators fall well short on production-level tasks, pinpointing the promising direction of agentic visual generation. We further identified two key challenges for this problem, blind search inject noise that can hurt generation, and agent must adapt to generator as it evolve capabilities. Our deliberately minimalist approach demonstrate evolving performance through co-training, setting a solid baseline for the problem.

AI USE DISCLOSURE

We used generative AI tools to construct research artifacts and evaluate model outputs. A frontier language model rewrote instantiated templates into naturalistic prompts for SEARCHGEN-20K; human annotators screened the resulting prompts and checked that their evaluation items were answerable from the prompt and cached search evidence (Appendix A.1). Gemini-3-Flash generated agentic search trajectories and provided scoring supervision for a finetuned Qwen3-VL-8B judge; that judge scored generated images against prompt-specific checklists and rubrics, including those used in the reported benchmark results. We compared the latter judge with human ratings on 500 prompt–image pairs (Appendix B.2). Image generators produced the images studied and training candidates. A generative AI assistant also helped draft and edit manuscript text and LaTeX, including this disclosure. The authors take responsibility for the final paper, its claims, and the released artifacts.

ETHICS STATEMENT

This work studies agentic visual generation using tasks synthesized from patterns and entities in production requests, human annotation, and archived image and web search results. Human annotators contributed to taxonomy labeling, prompt screening, and validation of the automated judge (Appendix A.1 and Appendix B.2). Improving fidelity to real-world people, symbols, and events can also make deceptive images easier to produce. Cached search results may contain personal information or copyrighted material and may reflect cultural or geographic biases that carry through to generated images. We screen prompts and search artifacts for culturally insensitive, personally sensitive, and unsafe content, and describe the intended research use and release safeguards in Appendix F.2. Users of the released resources should respect third-party rights and avoid deceptive impersonation or unauthorized reproduction; Appendix F discusses the remaining misinformation and bias risks.

REPRODUCIBILITY STATEMENT

Section 2.1 and Appendix A.1 describe task construction, annotations, and data splits. Section 2.2 and Appendix D.3 specify the archived-search replay and the evaluation procedure; the released task records, checklists, search corpus, and evaluation code support rerunning these steps with fixed evidence. Sections 3.1 and 3.2, Algorithm 1, and Appendix E document the agent protocol, training procedure, model settings, and compute. The experimental comparisons are in Section 4, with additional results and diagnostics in Appendix C.2 and Appendix D.5. Appendix F.1 identifies dependencies on third-party models and commercial APIs. Released assets are referred in the supplementary attachments.

REFERENCES

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.

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.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

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.

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.

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, pp. 10671–10684, 2022.

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.

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

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.

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.

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*, pp. 3043–3054, 2023.

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.

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.

Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-searcher: Reinforcing agentic search for image generation. *arXiv preprint arXiv:2603.28767*, 2026.

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.

Google Cloud. Imagen 3 Fast Generate 001: Model documentation. Vertex AI model documentation, 2024. URL https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/imagen/3-0-generate. Model ID: `imagen-3.0-fast-generate-001`.

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.

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*, pp. 3929–3938. PMLR, 2020.

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*, pp. 4754–4763, 2024.

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*, pp. 20411–20432, 2023.

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.

Imagen Team, Google, Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Lluis Castrejon, Kelvin Chan, Yichang Chen, Sander Dieleman, et al. Imagen 3, 2024. URL https://arxiv.org/abs/2408.07009.

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, pp. 1–43, 2023.

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.

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.

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*, pp. 7969–7992. Association for Computational Linguistics, 2023.

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*, pp. 15696–15707. PMLR, 2023.

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.

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, pp. 9459–9474, 2020.

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.

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.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=hFALpTb4fR.

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)*, pp. 9802–9822, 2023.

OpenAI. GPT-Image-2: Model documentation. OpenAI API model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-image-2. Model code: `gpt-image-2`.

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.

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*, pp. 22500–22510, 2023.

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.

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.

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*, pp. 8387–8396, 2024.

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. In *Forty-third International Conference on Machine Learning*.

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*, pp. 3060–3075, 2025a.

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)*, 2025b. Spotlight.

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)*, 2025c.

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)*, 2026a.

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*, 2026b.

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)*, 2026c.

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.

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.

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

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*, pp. 56704–56721. PMLR, 2024. URL https://proceedings.mlr.press/v235/yang24ai.html.

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.

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

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.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-Image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026. URL https://arxiv.org/abs/2605.10730.

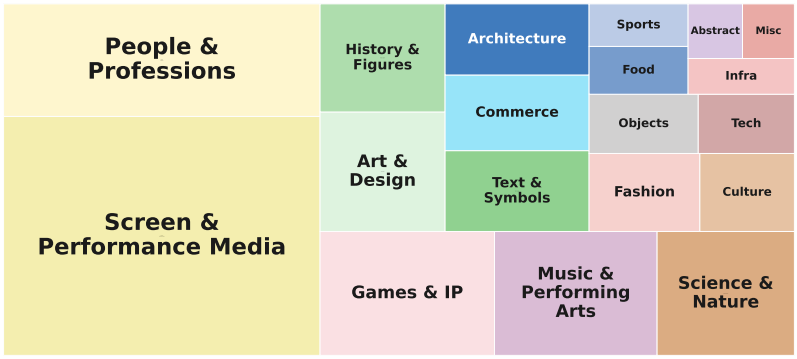


Figure 7: **SEARCHGEN-20K spans diverse, long-tailed domains.** Area reflects prompt counts across 22 domains; Figure 8 shows severity by domain and category.

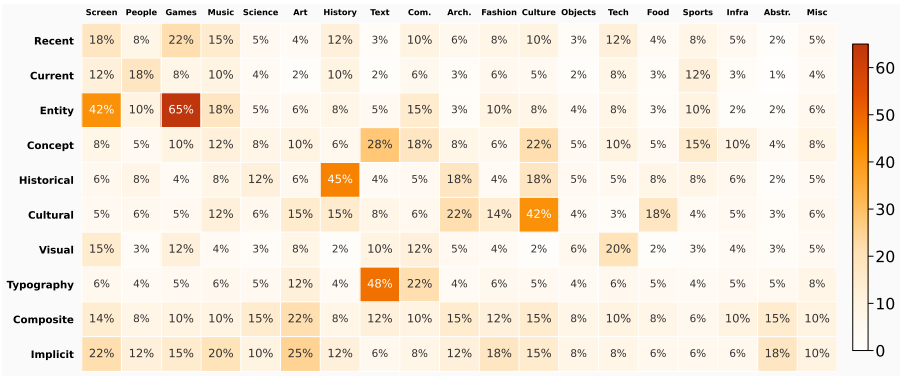


Figure 8: **The bottleneck is pervasive: every domain–category combination shows knowledge-driven degradation.** Heatmap summarizing cross-category structure between failure modes and domain categories in SEARCHGEN-20K. Degradation appears across domains and categories.

APPENDIX OVERVIEW

This appendix is organized to follow the evidence chain of the paper. §A.1 documents benchmark construction, domain coverage, 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. §C.2 expands the co-training stage, difficulty-tier, and generator-transfer comparisons and situates them within the broader generator landscape. §D.1–§D.5 provide the visual-generation paradigm contrast, workflow and reasoner I/O, evaluation protocol, formal capability-boundary definition, and paired retrieval-benefit and search-gate diagnostics. §E documents hardware and training/serving implementation details, including the co-training procedure in Algorithm 1 (§E.1). §??–§F summarize limitations, release safeguards, third-party/API caveats, and broader impact. §G contains the full Related Work discussion.

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 20,840 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 20,840 production 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. We retain long-tail entities to cover request types observed in production.

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 entity seeds, request patterns, relations, stylistic constraints, and visual or textual knowledge requirements distilled from the production prompts. 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. Gemini-3-Flash supplies scoring supervision for the finetuned Qwen3-VL-8B VLM judge; the latter assigns the scores reported in this paper. The judge input and nine-component aggregation are specified in Appendix D.3; the 500-pair human comparison is described 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 20,188 unique training prompts (including 128 held out for validation) and a disjoint **test** set of 749 unique prompts, matching Section 2.1; remaining non-test prompts support cached search construction, trajectory collection, and co-training rollouts. Unlike benchmarks that test compositionality within known concepts Li et al. (2024); Huang et al. (2023); Cho et al. (2023), 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. This can corrupt concepts the generator already knows.

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 the evaluated strata. Named entities and other shallow triggers are only loosely correlated with the evaluated NoSearch versus Search-Intensive distinction: 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). For Klein-4B-DPO on the TextualSearch stratum, both SFT AGENTand ADAPTED AGENTscore below the no-search baseline. The aggregate scores do not distinguish poor retrieval, integration noise, and difficulty rendering factual text as explanations. Such prompts require reasoning about typography, cultural composition, and implicit meaning that resists single-pass parameterization and search, motivating richer co-training cycles and search strategies.

Table 6: **The Search Bottleneck on our 749-prompt experimental subset – Full Nine-Component Breakdown.** Scores on a 0–100 scale (one decimal); higher is better. Open-weight generators drop sharply on the *Search-Intensive* set; declines among commercial systems vary. 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	46.3	50.0	46.7	37.0	55.6	16.0	45.8	62.8	58.0	32.1
	Flux.2-Klein-4B	46.9	51.7	48.7	36.5	58.1	10.5	46.5	64.5	60.7	29.4
	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

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 §2.1. 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 finetuned Qwen3-VL-8B judge achieves Spearman $\rho = 0.87$ with consolidated human labels on these 500 prompt–image pairs.

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 warrants caution when evaluating TextualSearch.

C Extended quantitative results

C.1 Full nine-component breakdown by stratum

Table 6 provides the nine-component evaluation for generators in our 749-prompt experimental subset, with 100 NoSearch and 649 Search-Intensive prompts. This supplements the six-component summary in the main paper (Table 4) with the full scoring breakdown.

C.2 Detailed co-training and transfer comparisons

The comparisons below expand the co-training results in Table 5. All changes are score-point differences on the same evaluation subsets. The NoSearch column denotes the 100-prompt subset, and search-intensive overall scores cover the remaining 649 prompts.

Contributions across stages and difficulty tiers. For Klein-4B, generator DPO adds 2.8 points to search-intensive quality and reasoner RFT adds 2.6, for a total increase of 5.4. The corresponding

Table 7: **Reasoner adaptation gains with base and DPO generators.** Each entry is RFT minus SFT quality with the same generator, calculated from Table 5. The RFT reasoner is trained with the DPO generator and then evaluated with both versions. Search-intensive overall scores average across the search-intensive prompts.

Generator	Version	NoSearch	Set I	Set II	Set III	Search overall
Klein-4B	Base	0.0	+0.1	+0.6	+0.3	+0.4
	DPO	+2.9	+2.3	+2.5	+2.7	+2.6
Bagel	Base	−0.1	+0.7	+0.7	+0.6	+0.7
	DPO	+1.9	+1.9	+2.4	+1.9	+2.1

Bagel gains are 1.3 and 2.1, totaling 3.4. From Phase 0 to Phase 2, Klein-4B gains 5.2, 4.4, and 6.2 points on Sets I, II, and III; Bagel gains 3.9, 2.7, and 3.4. Every tier benefits from each training stage, with the largest total gain on Set III for Klein-4B and Set I for Bagel.

The NoSearch trajectories complement these search-intensive gains. Klein-4B moves from 54.6 to 54.0 to 56.9 across the three phases, and Bagel moves from 52.6 to 52.4 to 54.3. Relative to Phase 0, the final systems improve by 2.3 and 1.7 points, respectively. Relative to generator-only inference with the DPO models, the final systems gain 7.0 and 5.5 points on NoSearch, alongside search-intensive gains of 6.8 and 4.4. Thus, both architectures benefit across the two prompt subsets.

Transfer to the base generator. Table 7 expresses the cross-check as the gain from replacing SFT with RFT while holding the generator fixed. With Klein-4B, the search-intensive RFT gain is 0.4 on the base generator and 2.6 on the DPO generator, a difference of 2.2 points. With Bagel, it is 0.7 and 2.1, a difference of 1.4. The RFT gain is larger with the DPO generator in every search-intensive tier and on the NoSearch subset for both architectures. This pattern connects the benefit of reasoner adaptation to the generator that supplies its image feedback.

Comparison with the frontier reasoner. With the same DPO generator, the adapted 8B reasoner scores 0.6 points above the Gemini-3-Flash reasoner on Klein-4B and 0.7 points above it on Bagel under the fixed automated judge. Its Set I–III score differences are $+0.4$, -0.3, and $+1.4$ for Klein-4B, and $+0.2$, $+0.3$, and $+1.5$ for Bagel. The largest numerical differences occur in the hardest tier for both architectures. NoSearch differences are $+1.2$ and $+1.1$, respectively. Training hardware and resource budgets are reported in Appendix E.

Position within the broader generator landscape. Table 6 provides the per-component breakdown for the 749-prompt experimental subset. Search-intensive overall scores range from 14.1 to 71.0 across those models. GPT-Image-2 scores 71.0, compared with 31.8 for our co-trained Klein-4B system, a gap of 39.2 points within this evaluation. These results place the co-training gains in the context of substantial remaining differences in absolute generation quality. Extending co-training to larger generators and improving evidence retrieval and integration are direct avenues for raising that quality; Appendix **??** discusses generator scaling and additional co-training cycles.

D Protocols and supplementary artifacts

D.1 Visual-generation paradigms

D.2 End-to-end workflow and agent I/O

The agent'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 §2.3.

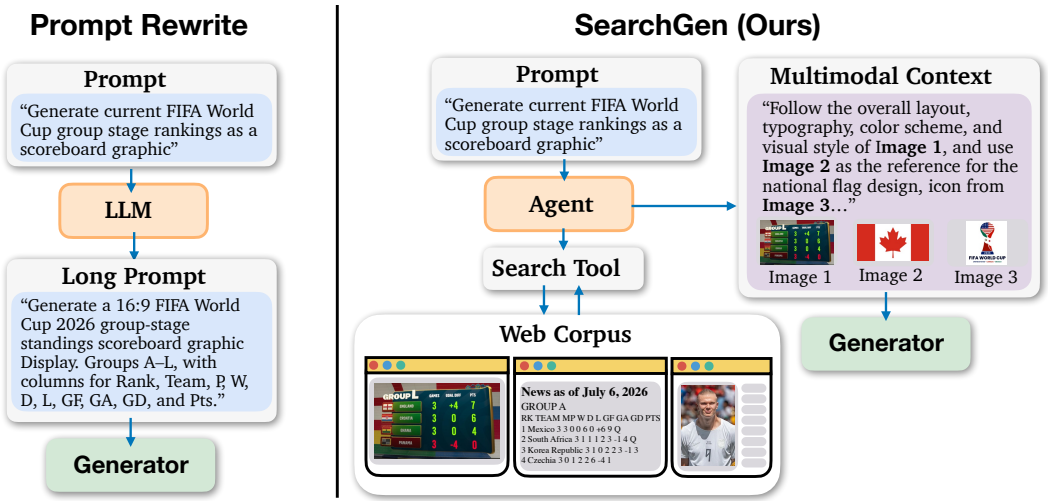


Figure 9: **Two paradigms for visual generation.** (*Left*) Prompt rewriting uses an LLM to expand the user query into a longer textual prompt passed to the generator. (*Right*) Our agent searches a web corpus for knowledge and visual references, organizes the retrieved evidence into multimodal context, and passes it to the generator.

End-to-end workflow (condensed). A user prompt enters; the agent runs gate → filter → 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.3).

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*. Its 120-degree spiral taper and tiered glass facade often render incorrectly; a reference image fixes their geometry.

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

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



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 agent 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 image shows the retrieved references and generation. Later boxes illustrate each stage with a distinct prompt.

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 render reliably; the prompt names no specific entity, event, or cultural tradition.

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

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

Stage 2 — Filter (SFT Task B). Once the gate fires, each image query returns up to five 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 turns selected references into a generation-ready prompt. The agent names what to borrow from each image ("following Image 1, render ...") so the generator uses the missing detail 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.3 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 checks that inform, but do not add to, 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.

Table 8: **Judge scoring components.** Each component is scored on $[0, 3]$. Components 1–2 are prompt-specific; components 3–8 are generic dimensions, with text rendering and physical plausibility scored when applicable; 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? Omit if no text is requested or rendered.
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. Omit for abstract or stylized images.
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.

Difficulty partition (Set I–III). The 649 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 finetuned Qwen3-VL-8B judge achieves Spearman $\rho = 0.87$ with human ratings on 500 prompt–image pairs (Appendix B.2).

D.4 Operational capability boundary

The main paper uses *capability boundary* as a descriptive concept. For a generator G_θ, prompt p from a fixed evaluation distribution $\mathcal{P}$, conditioning context c, and bounded quality measure $Q(G_\theta, p, c) \in [0, 1]$, let Q include the expectation over generation randomness. With a fixed retrieval protocol R and an optional prespecified adequacy threshold τ, we distinguish unaided quality from the additional value of retrieval:

$$q_0(p; \theta) = Q(G_\theta, p, \varnothing), \qquad \mathcal{C}_\theta(\tau) = \{p \in \mathcal{P} : q_0(p; \theta) \geq \tau\}, \tag{1}$$

$$\Delta_R(p; \theta) = Q(G_\theta, p, R(p)) - q_0(p; \theta). \tag{2}$$

The set $\mathcal{C}_\theta(\tau)$ describes requests handled adequately without retrieval under the chosen measure; Δ_R describes whether this particular retrieval protocol helps or hurts. A low or negative Δ_R may reflect poor retrieval, noisy references, or sufficient unaided ability. Training may change either quantity in either direction. Figure 6b reports the distribution of unaided quality q_0; the following subsection instead estimates Δ_R on matched prompts with a fixed retrieval package, and measures whether the agent's gate tracks the changed value.

Table 9: **Fixed-evidence retrieval benefit before and after generator DPO.** Means are on the 0–100 evaluation scale. s_R is the forced, fixed-package diagnostic and is not the gated agent score in Table 5. Search-Intensive contains 649 prompts; NoSearch contains 100.

		Base generator			DPO generator		
Generator	Stratum	$\bar{s}_0$	$\bar{s}_R$	$\bar{d}$	$\bar{s}_0$	$\bar{s}_R$	$\bar{d}$
Klein-4B	Search-Intensive	24.1	26.2	$+2.1$	25.1	28.9	$+3.8$
Klein-4B	NoSearch	51.0	48.5	-2.5	49.9	48.0	-1.9
Bagel	Search-Intensive	21.5	23.1	$+1.6$	22.4	24.5	$+2.1$
Bagel	NoSearch	49.3	47.1	-2.2	48.8	47.2	-1.6

D.5 Paired retrieval-benefit and search-gate diagnostics

Question and fixed-evidence intervention. The stagewise comparisons in Table 5 show that generator DPO and agent RFT improve the final system. To test whether they also change which requests benefit from search, we separately measure the two quantities in Appendix D.4: unaided quality and the incremental value of a *fixed* retrieval package. For each of the 749 test prompts, we build one package $R(p)$ from at most three queries in the archived search record and the first five cached results per query. A frozen SFT agent selects references and writes the integration specification once. We reuse that package, including the specification, for the base and DPO checkpoints of the same generator. The task's archived queries make this a diagnostic counterfactual, not an estimate of the deployed agent's end-to-end score; no test checklist or ground-truth answer is passed to the generator or agent. All 749 prompts have replayable packages in this analysis, and no live search is used.

Each prompt has four generation conditions: base and DPO generators, each without evidence and with $R(p)$. We draw four images per condition using matched seeds and score all outputs with the same evaluation prompt, checklist, rubric, and task-level reference evidence. We aggregate the applicable nine components as in Appendix D.3. To avoid a conditioning-dependent denominator in the knowledge analysis, we separately report $K(p) = (\mathrm{Checklist}(p) + \mathrm{Rubric}(p))/2$; visual-reference fidelity is analyzed only where it is applicable to both candidates under the same evaluation references. Let $s_0(p)$ and $s_R(p)$ be the per-prompt means of the four rescaled 0–100 scores, and let $d(p) = s_R(p) - s_0(p)$, the point-scale counterpart of Δ_R above. All intervals below are paired 95% bootstrap intervals from 10,000 resamples of prompts, rather than images. The five-point useful-retrieval threshold was selected on the 128-prompt validation split; the direction of the reported changes also holds at thresholds of zero, three, and ten points.

The benefit boundary changes in both directions. A mean difference cannot identify which prompts changed decision. We classify a prompt's fixed-retrieval benefit as useful when $d(p) > 5$, neutral when $-5 \le d(p) \le 5$, and harmful when $d(p) < -5$. Table 10 reports the complete base-to-DPO transition matrix on the 649 Search-Intensive prompts. For Klein-4B, 73 prompts leave the useful class ($65 + 8$) and 34 enter it ($29 + 5$), a net reduction of $39/649 = 6.0$ percentage points (paired 95% CI: $[2.9, 9.1]$). For Bagel, 66 leave ($59 + 7$) and 31 enter ($27 + 4$), a net reduction of $35/649 = 5.4$ points ($[2.5, 8.3]$). The pattern occurs in both evidence strata: useful retrieval changes from 190/386 to 166/386 on Klein-4B VisualSearch and from 124/263 to 109/263 on TextualSearch; the corresponding Bagel counts are 170 to 149 and 114 to 100. The mean value of retrieval nevertheless rises in the Search-Intensive stratum (Table 9): on the 241 Klein-4B prompts that remain useful, mean $d(p)$ increases from $+10.8$ to $+14.9$ points, and on the 218 analogous Bagel prompts it increases from $+10.2$ to $+12.5$. Thus the change is a redistribution of retrieval value across requests, not a uniform reduction in the need for evidence.

For the 73 Klein-4B prompts that move from useful to non-useful, the unaided knowledge score K improves by 7.1 points (95% CI $[5.6, 8.6]$), while its fixed-retrieval counterpart changes by only 0.6 ($[-0.4, 1.6]$); unaided Image Quality changes by 0.7 ($[-0.2, 1.6]$). The analogous Bagel changes

Table 10: **Per-prompt change in fixed-retrieval benefit on the 649 Search-Intensive requests.** Columns are the post-DPO classes; rows are base-generator classes. Each 3-by-3 block sums to 649. A useful-to-neutral or useful-to-harmful move is a request for which fixed retrieval no longer provides more than five points of benefit.

	Klein-4B after DPO			Bagel after DPO		
Base-generator benefit	Useful	Neutral	Harmful	Useful	Neutral	Harmful
Useful	241	65	8	218	59	7
Neutral	29	225	16	27	239	19
Harmful	5	25	35	4	20	56
Post-DPO totals	275	315	59	249	318	82

Table 11: **Search-gate decisions against fixed post-DPO retrieval benefit.** Useful means $d(p) > 5$; non-useful includes neutral and harmful outcomes. Entries are counts of prompts on which the agent searched. The final column reports gate-label agreement over the 649 Search-Intensive prompts.

Generator	Agent	Search when useful	Search when non-useful	Search on NoSearch	Gate agreement
Klein-4B	SFT	218/275	235/374	38/100	357/649 (55.0%)
Klein-4B	DPO-RFT	246/275	170/374	17/100	450/649 (69.3%)
Bagel	SFT	193/249	246/400	36/100	347/649 (53.5%)
Bagel	DPO-RFT	219/249	181/400	19/100	438/649 (67.5%)

on 66 prompts are $+6.2$ ($[4.9, 7.5]$), $+0.9$ ($[-0.1, 1.9]$), and $+0.5$ ($[-0.3, 1.3]$). These selected-subgroup differences are descriptive: the changed class is defined from the same outcomes. We therefore used two additional generation seeds for all 107 Klein-4B and 97 Bagel prompts whose usefulness classification changed in either direction; 87/107 and 79/97, respectively, retain the direction of their original switch. The improvement in unaided knowledge checks, together with stable searched performance on these prompts, argues against explaining the contraction solely as a failure to use references. It does not establish where that knowledge is represented.

The adapted gate tracks the post-DPO benefit. We next compare the gate decisions of the supervised SFT agent and the DPO-feedback RFT agent on the same prompts, before either agent sees the retrieved results. The fixed-package label $d(p) > 5$ is computed independently of both gate policies. Table 11 shows that the RFT agent is more likely to search when retrieval is useful, while searching less often when it is not. On Klein-4B Search-Intensive prompts, gate-label agreement rises from $357/649 = 55.0\%$ to $450/649 = 69.3\%$, a paired $+14.3$-point difference (95% CI $[10.1, 18.6]$). Bagel rises from $347/649 = 53.5\%$ to $438/649 = 67.5\%$, or $+14.0$ points ($[9.6, 18.3]$). On the separate 100-prompt NoSearch set, the number of search triggers falls from 38 to 17 for Klein-4B and from 36 to 19 for Bagel. Useful-retrieval recall rises from 218/275 to 246/275 on Klein-4B and from 193/249 to 219/249 on Bagel.

These results provide two distinct pieces of evidence absent from aggregate stagewise scores: generator training changes the per-prompt value of *the same* evidence, and the adapted agent's search decision better matches that changed value.

D.6 Search-protocol ablations

Matched intervention. We evaluate the Search-Intensive prompts with the same adapted agent and DPO generator for each architecture. Each condition reuses the archived candidate pool, at most three queries and five results per query, four matched generation seeds per prompt, and the same judge inputs and applicable nine-component scoring rule (Appendix D.3). The full protocol is compared with interventions. *No gate* executes a frozen query plan on requests the agent would skip, then applies the same filter and integration steps. *No filter* takes the top-ranked result while keeping the gate and integration steps. *No integration* keeps the selected reference but passes it with the original request, omitting the instructions that specify which attributes to borrow and which scene elements to create.

Table 12: **Search-protocol ablations on 649 Search-Intensive prompts.** Scores use the paper's 0–100 nine-component mean. Loss is the paired full-protocol advantage; brackets give prompt-level 95% bootstrap intervals. The full-protocol scores match Table 5.

	Klein-4B-DPO		Bagel-DPO	
Condition	Score	Loss [95% CI]	Score	Loss [95% CI]
Full protocol	**31.8**	—	**26.8**	—
No gate: force search	25.0	6.8 [5.5, 8.1]	20.8	6.0 [4.8, 7.2]
No filter: top result	26.4	5.4 [4.3, 6.5]	22.0	4.8 [3.8, 5.8]
No integration: same reference	27.6	4.2 [3.2, 5.2]	23.2	3.6 [2.7, 4.5]

Each step contributes to generation quality. Table 12 shows the largest loss when the gate is removed: 6.8 points on Klein-4B and 6.0 on Bagel. This result is measured within the Search-Intensive subset, where some requests still benefit from skipping retrieval. The full gate skips 233 Klein-4B and 249 Bagel requests; because the force-search intervention changes only those requests, the aggregate losses correspond to mean losses of 18.9 and 15.6 points on the skipped groups. The gate protects those requests from the quality loss induced by forced search.

Taking the top-ranked result in place of the filtered reference lowers the overall score by 5.4 points on Klein-4B and 4.8 on Bagel. The filter's choice therefore matters after the gate has selected a request for search. Keeping that choice but dropping the integration instruction costs 4.2 and 3.6 points relative to the full protocol.

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 Co-training procedure

Algorithm 1 gives the complete training sequence described in Section 3.2: supervised reasoner warm-start, generator preference training, and reasoner adaptation to the updated generator. The following subsections specify the training configurations and generator objective.

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 best-scoring Gemini-3-Flash trajectories $\mathcal{T}_\mathrm{sft} = \{(p_i, a_i^\mathrm{A}, a_i^\mathrm{B}, a_i^\mathrm{C})\}$ for Tasks A/B/C ▷ 20,080 trajectories
$R_{\phi_0} \leftarrow \arg\min_\phi \sum_{(p,a)\in\mathcal{T}_\mathrm{sft}} \mathcal{L}_\mathrm{SFT}(\phi; p, a)$ ▷ 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; operational quantities in Appendix D.4
    Construct preference pair: $x^w \leftarrow x_{\arg\max s_j},\ x^l \leftarrow x_{\arg\min s_j}$
  end for
  Update G_θ using the pairwise preference loss on $(x^w, x^l, \tilde{p})$
  ▷ Klein-4B flow-matching loss: Eq. 3; $\beta = 100$; Appendix E
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 $(G_{\theta'},\ R_{\phi'})$

E.2 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 13.

SFT data construction. Gemini-3-Flash runs the full gate–filter–integrate protocol on each training prompt, and the resulting images are scored by the VLM judge. For each prompt we keep the highest-scoring trajectory, giving 20,080 trajectories, a subset of the 96,848 released reasoning traces (Table 2). Tasks A, B, and C are the gate, filter, and integrate steps of the same trajectory, so the three training rows for a prompt share its search evidence and context. The data are 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.

E.3 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 14.

Pairwise objective for the flow-matching generator. Diffusion-DPO contrasts the policy and reference denoising errors of preferred and rejected images (Wallace et al., 2024). For Klein-4B, the corresponding Flow-Factory surrogate uses velocity-matching error. Let z^w and z^l be the latent

Table 13: **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× NVIDIA H20 (80 GB)
Training data	20,080 SFT trajectories
Wall-clock time	$<$4 hours

images selected under the same augmented condition c, and draw a shared noise tensor $\epsilon \sim \mathcal{N}(0, I)$ and flow time s for both arms. With $z_s^a = (1-s)z^a + s\epsilon$ and target velocity $u^a = \epsilon - z^a$ for $a \in \{w, l\}$, define the per-image error $E_m^a = d^{-1}\|v_m(z_s^a, s, c) - u^a\|_2^2$ for $m \in \{\theta, \text{ref}\}$, where d is the number of latent elements. The optimized pairwise surrogate is

$$\mathcal{L}_{\text{FM-DPO}}(\theta) = -\mathbb{E}_{(c,z^w,z^l),s,\epsilon}\left[\log \text{sigmoid}\left(-\frac{\beta}{2}\left[(E_\theta^w - E_{\text{ref}}^w) - (E_\theta^l - E_{\text{ref}}^l)\right]\right)\right]. \quad (3)$$

The reference is frozen during this update. The factor $1/2$ follows the Flow-Factory pairwise loss implementation[1]; $\beta = 100$ is the configured scale. This is a velocity-error *surrogate* for the flow model, rather than an exact output-density ratio. Wallace et al.'s original diffusion objective uses noise-prediction errors and a timestep/weighting scale; Eq. 3 states the objective appropriate to the flow-matching implementation.

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. For DPO candidate scoring, the Qwen3-VL-8B judge uses greedy decoding, a maximum of 4,096 output tokens, and a maximum pixel budget of 262,144 per image, with up to 24 concurrent requests. Final benchmark evaluation also uses greedy decoding.

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.

[1] https://github.com/X-GenGroup/Flow-Factory/blob/main/src/flow_factory/trainers/common/dpo_objective.py

Table 14: **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 15: **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

E.4 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× H20, same hyperparameters), with the key difference that the training data consists of filtered trajectories: only reasoner rollouts with positive group-relative advantage are retained for training; this means they scored above the sampled group mean, without directly comparing search to a no-search output. The full RFT cycle completes in approximately $4 \times 8 = 32$ GPU-hours.

E.5 Total compute budget

Table 15 summarizes end-to-end training compute (Phase 1 GPU-hours are 8 GPUs $\times$24 h wall-clock). The VLM judge runs on a separate GPU, excluded from training compute.

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 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 20,937 unique 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 visual AI that can serve users across cultures, including those poorly 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 AI naturalness evaluation dimension that measures detectable artifacts, and our released assets also include trained SearchGen checkpoints and cached search results. We encourage provenance tracking and watermarking for downstream use.
- **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.

F.1 THIRD-PARTY MODELS, DATA ACCESS, AND TERMS OF USE

Open-weight generators. Bagel Deng et al. (2025); FLUX.2 Klein checkpoints Black Forest Labs (2026); Qwen-Image Wu et al. (2025). **Commercial APIs.** Qwen-Image-2 Zhao et al. (2026); Gem-

ini developer endpoints (Gemini 2.5 Flash Image, Gemini 3 Pro Image; consumer names Nano Banana / Nano Banana Pro in some regions) Google DeepMind (2025); ByteDance Jimeng ByteDance (2025); SeedDream 4.0 ByteDance Seed Team (2025); Imagen 3 Fast Imagen Team, Google et al. (2024); Google Cloud (2024). 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; the public release includes prompts, generated outputs, trained SearchGen checkpoints, code, and the complete cached search corpus; third-party base-model weights remain with their original providers.

F.2 RESPONSIBLE RELEASE AND SAFEGUARDS

Release scope. SEARCHGEN-20K and SEARCHGEN-BENCH are intended for research evaluation of knowledge-intensive visual generation. The full public release will include prompts, annotations, evaluation checklists, rubrics, search queries, reasoning trajectories, generated images, trained SearchGen checkpoints, code, and the complete SEARCHGEN-CORPUS-1M archive with cached image and web downloads. Indices and replay code support searches without live APIs.

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.

G FULL RELATED WORK

Search-augmented generation. Retrieval-augmented generation (RAG) (Lewis et al., 2020) and REALM (Guu et al., 2020) established the paradigm of grounding language models in external documents at inference time. Atlas (Izacard et al., 2023) and REPLUG (Shi et al., 2023) 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 (Chen et al., 2023) conditions diffusion on searched image–text pairs, RDM (Blattmann et al., 2022) augments generation with nearest-neighbor features, KNN-Diffusion (Ashual & Wolf, 2023) queries a large image database at generation time, Gen-Searcher (Feng et al., 2026) trains an agentic searcher via SFT and RL while keeping the image generator frozen, and IP-Adapter (Ye et al., 2023) injects visual exemplars through cross-attention adapters. Subject-driven methods such as DreamBooth (Ruiz et al., 2023) and Textual Inversion (Gal et al., 2023) internalize specific concepts via per-concept optimization; BLIP-Diffusion (Li et al., 2023) 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. In contrast, we study real user requests from production AIGC systems, identify real-world failure mode taxonomy, and contribute datasets and replayable search harness with this real-world diversity. Our analysis (§2) shows that indiscriminate search actively degrades a substantial fraction of prompts, making the *decision to search* itself a first-order design variable. Our approach thus proposes to co-evolve the joint system of agentic reasoner and image generator, laying the foundation for recursive self-improvement in visual generation.

Selective and adaptive retrieval. The question of *when* to search has been studied extensively in text. Self-RAG (Asai et al., 2024) trains a language model to emit retrieval-control and critique tokens. FLARE (Jiang et al., 2023) triggers retrieval when predictive confidence drops. Toolformer (Schick et al., 2023) learns when to invoke external APIs including search. CRAG (Yan et al., 2024) evaluates and corrects retrieval quality post-hoc. Adaptive-RAG (Jeong et al., 2024) routes queries to different retrieval strategies based on complexity. Mallen et al. (Mallen et al., 2023) and Kandpal et al. (Kandpal et al., 2023) show that popular entities are often better served by parametric knowledge while rare entities benefit from retrieval, motivating a generator-dependent

comparison of unaided quality and retrieval benefit. These methods establish selective retrieval for text, but visual search failure is structurally different: it manifests as spatial distortion, identity blending, and stylistic contamination (§2.3), failure modes that have no text-domain analogue and require modality-specific triage and integration. Our recipe instead trains the generator and then finetunes the reasoner against that updated generator, allowing us to evaluate the effect of one generator–reasoner training cycle.

Tool-augmented and agentic image generation. GenAgent (Jiang et al., 2026), RPG-DiffusionMaster (Yang et al., 2024), LLM-grounded Diffusion (Lian et al., 2024), and Iter-Comp (Zhang et al., 2024) use LLM-based planning to decompose complex prompts, improving compositional generation within the generator's existing knowledge. Instruct-Imagen (Hu et al., 2024) supports multi-modal instructions including style exemplars. RationalRewards (Wang et al., 2026b) 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 (Li et al., 2024) tests prompt adherence, T2I-CompBench (Huang et al., 2023) evaluates spatial reasoning and attribute binding, DALL-Eval (Cho et al., 2023) measures object relationships, and HEIM (Lee et al., 2023) provides holistic multi-dimensional evaluation. VLM-based automated judges such as TIFA (Hu et al., 2023) and human-grounded auto-metric suites such as Gecko (Wiles et al., 2024) 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 (Yuan et al., 2024) show that a model can judge its own outputs to generate training signal, improving over successive iterations without external reward. SPIN (Chen et al., 2024) frames self-improvement as self-play, iteratively distinguishing model outputs from ground truth. Our two-stage recipe applies preference training to the generator using search-augmented inputs, then finetunes the reasoner against the updated generator. The sequential pairing of these two components distinguishes the system setup; the present experiments measure one pass through each stage.