AGENTIC IMAGE GENERATION IN THE WILD

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic image generation pairs an image generator with a search agent that retrieves references from the web, extending generation beyond the generator's training data. Real-world requests, however, often concern entities that appeared only days ago. We call this setting Image Generation in the Wild. In this setting, the web offers limited visual evidence, search results are crowded with easily confused entities, and the target is rarely shown on its own. To study this setting, we build an automated pipeline that synthesizes prompts from recent headlines and keeps only those whose real search results are highly confusing, yielding requests that are hard to solve yet natural to ask. With it, we construct Savanna, a benchmark of 500 prompts across 16 subcategories. On Savanna, even harness frameworks built on the frontier models retrieve a correct reference for only about half of the prompts. We then propose Locate-Corroborate-Isolate, a paradigm in which the agent first pins down the target's identity through text search, then verifies candidate images against multiple sources, and finally crops a clean reference for the generator. Trained on 15K trajectories that follow this paradigm, Ranger-8B achieves a search score of 67.2, outperforming harnesses equipped with frontier models. It also achieves the best generation quality with every generator we test, and surpasses prior agentic methods on other image generation benchmarks. We will release the pipeline, benchmark, and training data.

1 INTRODUCTION

Generative models (Wu et al., 2025; Google, 2025b) have made substantial progress in generating high-quality images from natural-language prompts. Agentic image generation (Feng et al., 2026; Chen et al., 2026b) further extends these capabilities by equipping vision-language model (VLM) agents with web search tools to retrieve textual and visual references, improving performance on tasks that require reasoning or knowledge of uncommon entities (Niu et al., 2025; He et al., 2026). Despite this progress, current systems still struggle in realistic settings, which we refer to as **Image Generation in the Wild**. We identify three major challenges in this setting (Fig. 1). **Temporal novelty:** users may request images of entities shortly after their introduction. Unlike an established uncommon entity, a newly introduced one lacks a stable visual record, making its appearance harder to confirm. **Semantic crowding:** web searches often return entities that are easily confused with the target. For example, a search for a newly released car may also return its earlier versions, leading agents to select incorrect references. **Spatial entanglement:** available images may show the target only together with other objects or against a distracting background, which interferes with downstream generation.

To study and address these challenges, we introduce an end-to-end framework for agentic image generation in the wild, spanning data synthesis, evaluation, and model training. We first build an automated pipeline that turns recent headlines into image generation prompts that exhibit these challenges. It starts by collecting recent news and related reports into *world snapshots*, which serve as the source of newly introduced entities. To mimic the web that a time-sensitive request faces, all later searches are restricted to results published within three days of each article. The pipeline then extracts target entities from each snapshot, either from the text or from objects that appear only in the retrieved images. Next, it measures the *retrieval confusion* of each entity and keeps only entities whose search results are sufficiently confusing. A two-stage VLM verification then ensures that each remaining entity points to a single, unambiguous answer. Finally, each entity is written into a prompt with a ground-truth reference and evaluation rubrics. Together, these designs reproduce

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.

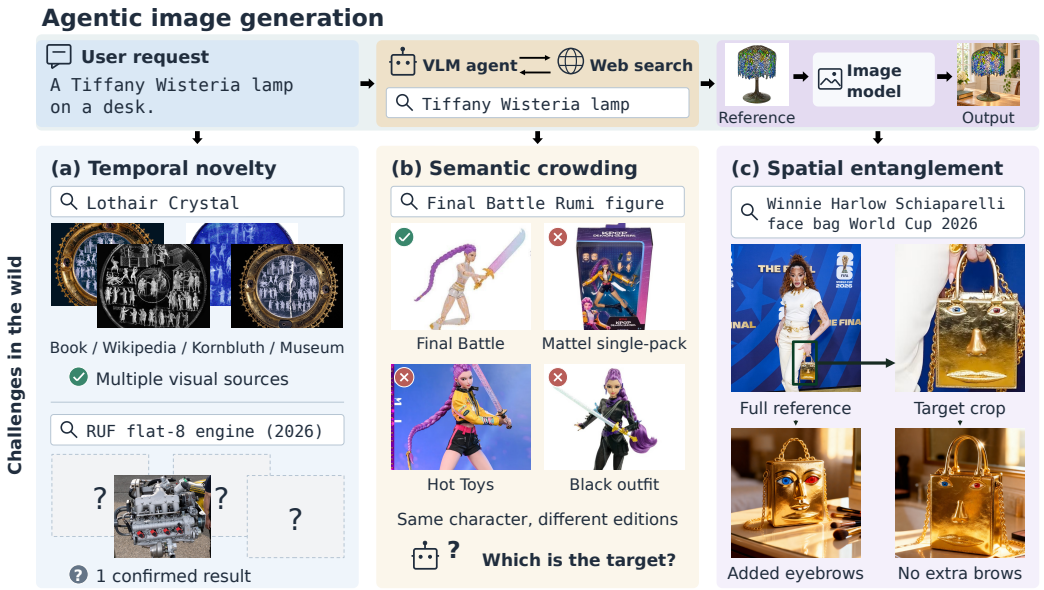


Figure 1: **Agentic image generation and its challenges in the wild.** Top: the agentic image generation workflow. Bottom: (a) temporal novelty, (b) semantic crowding, and (c) spatial entanglement.

temporal novelty, semantic crowding, and spatial entanglement in the resulting prompts, which still read like natural user requests and can be synthesized continually as long as news keeps arriving.

Using this pipeline, we construct **Savanna**, a benchmark of 500 prompts spanning 16 subcategories in five domains. We evaluate retrieval and generation separately: the *search score* measures whether the agent retrieves correct visual references for the target, and the *generation score* measures whether the final image faithfully depicts the target in the requested context. From proprietary generators to agentic post-trained models, all systems struggle on Savanna. Even harness frameworks (Ren et al., 2026; Ye et al., 2026) built on the latest frontier models such as GPT-6-Astra (OpenAI, 2026b) and Claude-Opus-5 (Anthropic, 2026b) reach a search score of at most 51.5.

These results suggest that simply collecting search results and passing them to the generator is insufficient in the wild. We therefore propose the *Locate-Corroborate-Isolate* paradigm. First, in the Locate phase, the agent uses text search and web fetching to establish the target's identity before searching for images, compensating for the missing visual record. Next, in the Corroborate phase, the agent carefully checks each candidate image against the previously gathered facts and other candidates, addressing semantic crowding. Finally, in the Isolate phase, the agent crops the image when necessary into a clean reference, addressing spatial entanglement. To teach this paradigm, we reuse our pipeline to synthesize training prompts and use a teacher model to solve them following the paradigm. We finally keep 15K trajectories whose final references match the ground truth. Fine-tuned on these trajectories, **Ranger-8B** reaches a search score of 67.2, 15.7 points above the strongest baseline and more than 20 points above agents with the same backbone. It also achieves the highest generation score with every downstream generator we test, and outperforms prior agentic methods on Mind-Bench (He et al., 2026) and KnowGen (Feng et al., 2026). Further analysis validates the effectiveness of our design in addressing these challenges.

2 Savanna: Is Image Generation Ready for the Real World?

This section first formulates agentic image generation and the challenges it faces in the wild (Section 2.1). We then describe how we synthesize queries that exhibit these challenges (Section 2.2) and how we curate and evaluate Savanna (Section 2.3).

2.1 TASK FORMULATION

Agentic image generation. An image generator can only depict entities it has learned, yet users often ask for entities it has never seen. Agentic image generation closes this gap by placing an agent π before the generator. Given a user prompt p that asks for a newly introduced entity e, the agent searches the web $\mathcal{W}$ for textual and visual knowledge about e. It then outputs a refined prompt $\hat{p}$ and a set of reference images $\mathcal{I}_{\text{ref}}$, from which a generator G produces the final image I_{gen}:

$$(\hat{p}, \mathcal{I}_{\text{ref}}) = \pi(p;\, \mathcal{W}), \qquad I_{\text{gen}} = G(\hat{p}, \mathcal{I}_{\text{ref}}). \tag{1}$$

The Call of the Wild. In real-world scenarios, users often ask for entities that have only just appeared, and the agent can only search the web as it is at that moment. Let τ denote the time a prompt is issued, and $\mathcal{I}_e$ the images on $\mathcal{W}$ that depict e. We observe three core challenges in this setting, illustrated in Fig. 1.

- **Temporal novelty.** When e is introduced shortly before τ, the web has had little time to catch up. Its images may not yet be indexed by search engines, so a direct image search for e often misses them. Even when some are found, $\mathcal{I}_e$ is too small to form a visual consensus, making it hard to tell which retrieved image actually depicts e (Fig. 1a).
- **Semantic crowding.** A newly introduced entity rarely stands alone in search results. Instead, its results are mixed with other entities that share its name or belong to the same product line. For example, a search for a club's new home kit also returns last season's kit, and a search for a new film returns posters of an older film with the same title. Such results easily mislead the agent into selecting the wrong entity (Fig. 1b).
- **Spatial entanglement.** Many entities are photographed only together with other objects or against a distracting background, such as one product among several on a launch stage. Even an image that correctly depicts e may then be unsuitable as a reference, since the generator tends to reproduce everything it contains (Fig. 1c).

Together, these challenges make a correct and clean reference for e hard to obtain.

2.2 FROM HEADLINES TO HARD PROMPTS

Studying image generation in the wild calls for a steady supply of prompts that exhibit these challenges, both to evaluate agents and to train them. The most direct approach is to collect and filter real user prompts, but this requires access to a large and steady stream of user requests. Can similar prompts instead be constructed automatically from scratch? Our answer is an automated pipeline that turns recent headlines into hard prompts, illustrated in Fig. 2 and described below.

World snapshot collection. The very first step is to collect recent news articles. We predefine a set of domain-specific meta queries and use them to search for news day by day over a given time range, recording the title, snippet, source, and publication date of each article. Next, we enrich the context of each article by fetching its full text and running another text search on its title to gather related reports from other sources. The article and this supplementary context together form a *world snapshot*: a record of what the web says about an event shortly after it happens. Specifically, we treat the news date as the issue time τ and restrict all subsequent text or image searches to results published no later than three days after τ, thereby forming a web environment with temporal novelty. Implementation details are presented in Appendix D.

Entity extraction. Given a world snapshot, we can now derive the target entity e that a prompt p will ask for. We extract entities in two ways, called text-oriented and image-oriented, respectively. In *text-oriented* extraction, an LLM directly extracts an entity from the world snapshot (*e.g.,* the Schiaparelli T-shirt that Winnie Harlow wore to the 2026 World Cup). We require the entity to be visually identifiable and to carry distinctive attributes, so that it is worth retrieving and can be used to judge whether a generated image is correct. However, an entity reported in the text is usually shown as the main subject in news photos, which naturally provide clean references. To introduce spatial entanglement, we further perform *image-oriented* extraction: a VLM checks the images retrieved for a text-oriented entity and picks another entity that is visible in these images but not mentioned in the text (*e.g.,* the face-shaped bag in Fig. 2). Finally, each extracted entity is written as an image search query u and passed to the next step.

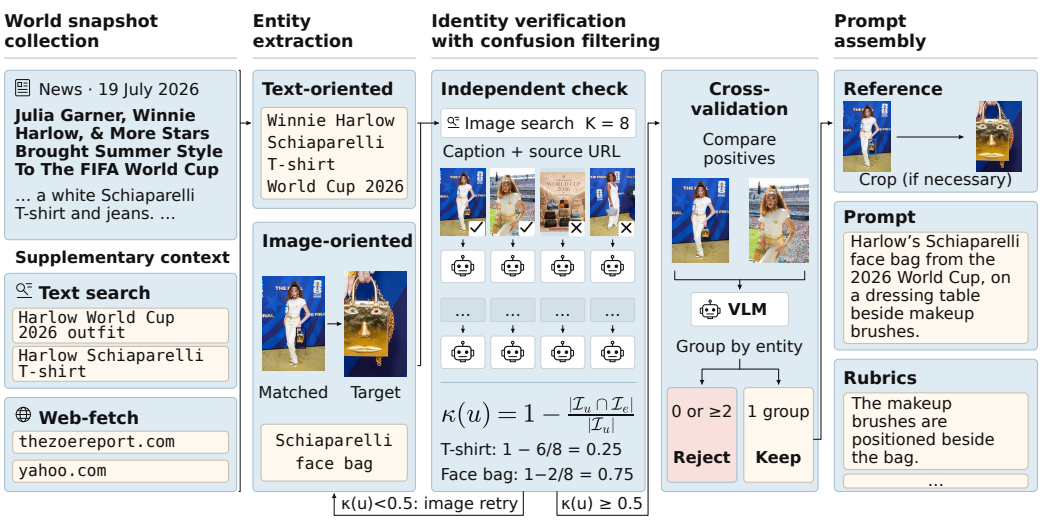


Figure 2: **Prompt synthesis pipeline**, illustrated with real retrieval results. The text-oriented T-shirt query is too easy ($\kappa = 0.25$), so image-oriented extraction proposes the face bag ($\kappa = 0.75$).

Identity verification with confusion filtering. Given the search queries, we next identify which retrieved images truly depict each target entity, filter out queries that are too easy, and ensure that each remaining query has exactly one correct answer. Specifically, for each query u we retrieve a set $\mathcal{I}_u$ of $K = 8$ images. These images first go through the *independent check* phase, where a VLM judges each image separately, together with its caption and source URL. We then define the *retrieval confusion* $\kappa(u)$ as the fraction of results that do not depict e:

$$\kappa(u) = 1 - \frac{|\mathcal{I}_u \cap \mathcal{I}_e|}{|\mathcal{I}_u|}. \tag{2}$$

We keep only queries with $\kappa(u) \geq 0.5$, so that at least half of the search results show a wrong entity, thereby ensuring semantic crowding around each target entity. Text-oriented queries below the threshold are considered too easy and are sent back to the previous phase for image-oriented extraction. For the remaining queries, the images in $\mathcal{I}_u \cap \mathcal{I}_e$ are sent to the *cross-validation* phase, where a VLM compares them jointly and groups them by whether they show the same entity. We discard queries that match no group, which have no answer, and those that match two or more groups, which are ambiguous. The images in the single remaining group form the candidate set $\mathcal{I}_{\text{cand}}$, from which the reference images are later selected.

Prompt assembly. At last, each remaining query is turned into an image generation prompt, paired with a ground-truth reference and rubrics for evaluation. First, a VLM selects the image in $\mathcal{I}_{\text{cand}}$ that shows e most clearly, and crops e from it when necessary, forming the reference images $\mathcal{I}_{\text{ref}}$. Then, a VLM expands e into a prompt p that places e in a new scene and describes its interaction with the surroundings, without revealing its appearance. Finally, the scene information in p is extracted as rubrics for the subsequent evaluation.

Together, these steps produce prompts that concern newly introduced entities, are difficult under real search conditions, and have a single verified answer, while remaining close to what a user would actually ask. The pipeline also can keep producing prompts as long as news keeps arriving.

2.3 Benchmark Curation

Following this pipeline, we build our benchmark, **Savanna**. We select five domains that users frequently ask about, namely Entertainment, Sports, Fashion, Products, and Culture, covering 16 subcategories. For each subcategory, we design several meta queries, for a total of 50 (fully presented in Appendix E). We restrict news articles to July 2026, yielding 500 prompts, 399 from text-oriented and 101 from image-oriented extraction. Figure 3 shows their distribution over domains and subcat-

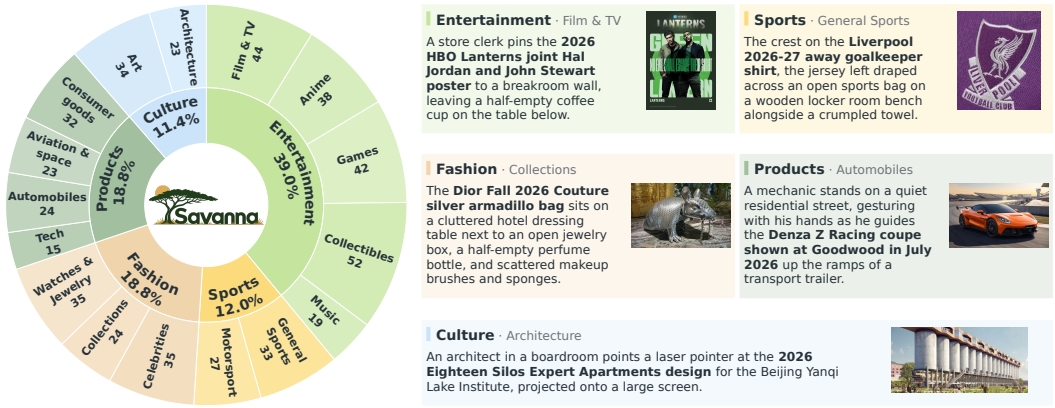


Figure 3: **Overview of Savanna.** The rings show the five domains (inner) and 16 subcategories (outer). Each domain has one example prompt with its reference image.

egories. Finally, three PhD-level annotators independently review every sample, ensuring that it has exactly one correct answer and the reference correctly corresponds to the target entity.

Metrics. We evaluate agents with two scores. The *search score* is the fraction of prompts whose retrieved references match the ground-truth reference of e. The *generation score* rates I_{gen} mainly on whether e is drawn consistently with the ground-truth reference, and partly on whether the context in p satisfies the rubrics. Both scores are evaluated by a VLM. Details are given in Appendix F.

3 Ranger: Leading Image Generation Through the Wild

Each challenge in the real world calls for a dedicated step in the search. Accordingly, we propose a reasoning paradigm that *locates* the identity of e from text before searching for images, *corroborates* candidate images against their sources and against each other, and *isolates* the verified target into a clean reference (Section 3.1). Following this paradigm, we train **Ranger-8B**, a VLM agent learned from teacher trajectories on synthesized prompts (Section 3.2).

3.1 Locate, Corroborate, Isolate: Surviving the Wild in Three Steps

To let the agent gather knowledge from the web, we first build an agent environment with four tools: `text_search` and `web_fetch` for textual evidence, `image_search` for candidate images, and `crop_image` for cutting a region out of a retrieved image. On top of this environment, we design a three-step reasoning process. Tool implementations are presented in Appendix D.

Locate. A natural first move is to search for images of e directly with the prompt. For an entity that has only just appeared, however, this is unreliable: its images may not yet be indexed, and the few that are found offer no visual consensus for telling which of them shows e. The prompt offers little help with either problem, as it names e but says little about what sets it apart. The agent therefore turns to text first, since textual reports can supply exactly these details. It uses `text_search`, and `web_fetch` when snippets are insufficient, to gather what is known about e, such as its maker, release event, and distinguishing features. These facts serve two purposes: they yield an image search query more precise than the prompt, and they provide the criteria for judging candidates in the next step. In Fig. 4, the agent learns from the official release that the 2026 Zero Sugar bottle has a black cap and an updated label before issuing its first image search.

Corroborate. A precise query narrows the search, but does not clean it up. The results remain crowded with entities that share the name or product line of e, each of which can look plausible on its own. Rather than trusting any single image, the agent corroborates each candidate in two ways. First, it checks the candidate *across modalities*: besides the pixels, it reads the accompanying text, such as the image title and source page, and compares both with the facts from Locate, rejecting

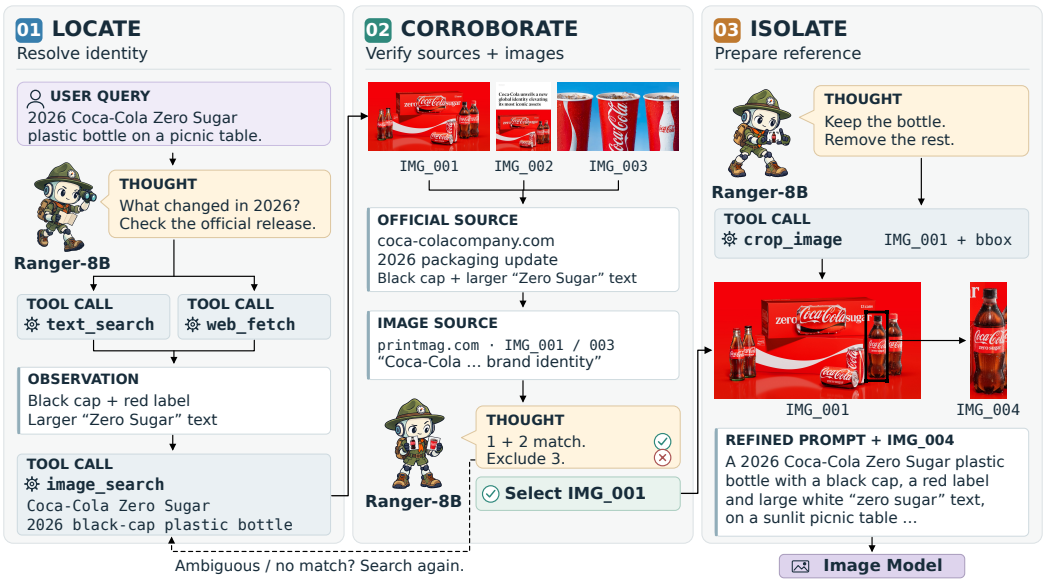


Figure 4: **The Locate–Corroborate–Isolate paradigm.** The agent identifies e from text, verifies candidates against sources and each other, then crops a clean reference for the generator.

candidates that show a different version, colorway, or member of the same product line. Second, when several candidates remain, it checks them *across images*, looking for a group that shows the same entity. Such agreement, when present, adds confidence that no wrong entity has slipped through. If no candidate remains or several groups are evenly matched, the agent returns to Locate to gather more context about e, and then searches again with a more distinctive query.

Isolate. A verified image shows the right entity, but not necessarily that entity alone. It may also contain other objects or a distracting background. Given such a reference, the generator may confuse which object the prompt refers to, or carry the extra content into its output, degrading the generated image. The last step therefore disentangles e from its surroundings. The agent selects the verified image whose view best matches p, and when e shares the frame with other objects, calls `crop_image` with a bounding box around e, which returns a new, cropped image with its own ID. In Fig. 4, the agent crops the bottle out of a verified product lineup that also includes cans and glass bottles, and passes the crop to the generator together with $\hat{p}$.

3.2 Learning from Teacher Trajectories

The three stages require behaviors that current open models rarely show. We first collect teacher trajectories and train Ranger-8B.

Trajectory collection. We run the pipeline in Section 2.2 on news from January to June 2026 to obtain training prompts, which are disjoint in time from Savanna, avoiding data leakage. We then adopt Gemini-3.1-pro (Google, 2026a) as the teacher model to solve each prompt in the same agent environment, with the proposed reasoning paradigm. We keep only trajectories whose final reference matches the ground truth, and discard those whose $\hat{p}$ drops conditions of p or that repeat searches without gaining new information. The remaining 15,119 trajectories form **Ranger-15K**.

Training details. Utilizing these trajectories, we train Ranger-8B on Ranger-15K using 8 NVIDIA H20 GPUs, with Qwen3-VL-8B-Instruct (Bai et al., 2025) as the base model. The learning rate is set to 1×10^{-5} with a cosine schedule, the batch size to 32, and the maximum context length to 40K tokens. For both trajectory collection and inference, the agent is limited to 8 tool calls, and each `image_search` returns at most 5 images.

Table 1: Search and generation results on Savanna by domain. Bold/underline mark best/second-best scores. * indicates methods with a stronger model. Qwen-IE-2511 denotes Qwen-Image-Edit-2511.

Method	Generator	Entertainment		Sports		Fashion		Products		Culture		All	
		Search	Gen.	Search	Gen.	Search	Gen.	Search	Gen.	Search	Gen.	Search	Gen.
Proprietary Generative Systems													
–	GPT-Image-2	–	23.4	–	29.5	–	25.3	–	29.9	–	23.7	–	26.4
–	Nano Banana Pro	–	25.9	–	30.5	–	23.4	–	28.5	–	22.4	–	26.1
–	Nano Banana 2	–	33.6	–	37.7	–	31.3	–	39.2	–	32.6	–	34.9
Harness Frameworks													
Mind-Brush GPT-5.1	Qwen-IE-2511	39.0	31.4	38.3	26.8	29.8	23.8	45.7	33.1	26.3	32.6	35.8	29.5
	Nano Banana Pro		47.6		40.6		36.4		46.7		39.5		42.2
SCOPE GPT-5.4	Qwen-IE-2511	49.2	37.1	38.3	22.2	35.1	27.9	56.4	36.4	26.3	27.4	41.1	30.2
	Nano Banana Pro		49.4		42.7		44.5		50.8		38.0		45.1
SCOPE* GPT-6-Astra	Qwen-IE-2511	55.9	39.5	48.3	37.3	39.4	30.1	59.6	39.9	54.4	39.5	51.5	37.3
	Nano Banana Pro		47.3		48.5		41.3		50.8		50.0		47.6
GenClaw Claude-Opus-4.6	Qwen-IE-2511	51.8	37.7	40.0	33.6	36.2	28.9	55.3	37.2	42.1	33.9	45.1	34.2
	Nano Banana Pro		46.4		45.8		41.2		47.0		40.9		44.2
GenClaw* Claude-Opus-5	Qwen-IE-2511	55.4	39.3	43.3	35.1	38.3	29.7	59.6	39.5	52.6	38.7	49.8	36.5
	Nano Banana Pro		47.0		45.3		40.7		50.3		48.9		46.4
Unified Multimodal Models													
Unify-Agent BAGEL-14B	BAGEL	20.0	21.1	18.3	23.2	22.3	21.9	30.9	22.9	24.6	23.2	23.2	22.5
	Qwen-IE-2511		23.5		24.9		24.5		24.9		26.8		24.9
	Nano Banana Pro		30.4		31.9		29.9		38.6		27.2		31.6
Agentic Post-trained Models													
Gen-Searcher Qwen3-VL-8B	Qwen-IE-2511	46.7	35.5	36.7	36.1	36.2	31.9	48.9	38.2	49.1	47.7	43.5	37.9
	Nano Banana Pro		44.9		48.7		44.1		45.8		52.8		47.3
GenEvolve Qwen3-VL-8B	Qwen-IE-2511	45.1	41.3	36.7	28.6	29.8	29.3	42.6	33.0	33.3	38.9	37.5	34.2
	Nano Banana Pro		48.2		43.6		36.7		43.3		41.7		42.7
Ranger Qwen3-VL-8B	Qwen-IE-2511	**71.3**	**50.0**	**66.7**	**39.6**	**71.3**	**48.8**	**61.7**	**45.3**	**64.9**	**50.3**	**67.2**	**46.8**
	Nano Banana Pro		**58.6**		**59.1**		**65.5**		**52.6**		**58.7**		**58.9**

4 EXPERIMENTS

4.1 EVALUATION SETUP

Baselines. We compare Ranger-8B with four families of baselines: (1) Proprietary generative systems, GPT-Image-2 (OpenAI, 2026c), Nano Banana Pro (Google, 2025b), and Nano Banana 2 (Google, 2026b), which we call through their official APIs with built-in web search enabled; (2) Harness frameworks, Mind-Brush (He et al., 2026), SCOPE (Ren et al., 2026), and GenClaw (Ye et al., 2026), which equip proprietary LLMs (GPT-5.1 (OpenAI, 2025a), GPT-5.4 (OpenAI, 2026a), and Claude-Opus-4.6 (Anthropic, 2026a), respectively) with search tools, where SCOPE and GenClaw are additionally run with the stronger GPT-6-Astra (OpenAI, 2026b) and Claude-Opus-5 (Anthropic, 2026b); (3) Unified multimodal models, represented by Unify-Agent (Chen et al., 2026a); and (4) Agentic post-trained models, Gen-Searcher (Feng et al., 2026) and GenEvolve (Chen et al., 2026b), which share the same backbone with Ranger.

Benchmarks. We mainly evaluate on Savanna with two downstream generators, Qwen-Image-Edit-2511 (Wu et al., 2025) and Nano Banana Pro (Google, 2025b). To test generalization beyond Savanna, we further evaluate on two representative benchmarks, Mind-Bench (He et al., 2026) and KnowGen (Feng et al., 2026). Benchmark and evaluation details are given in Appendix F.

4.2 MAIN RESULTS ON SAVANNA

Existing methods struggle in the real-world setting. As shown in Table 1, no baseline finds the right reference reliably. Even harness frameworks equipped with the latest frontier LLMs (*e.g.,* SCOPE with GPT-6-Astra and GenClaw with Claude-Opus-5) pass only about half of the prompts.

Table 2: Results on Mind-Bench and KnowGen. Bold/underline mark best/second-best scores among open-source methods.

Method	Generator	Mind-Bench			KnowGen				
		Knowledge	Reasoning	Overall	Visual cor.	Text acc.	Faith.	Aesth.	K-Score
Proprietary Generative Systems									
–	GPT-Image-1	15.6	17.6	17.0	20.18	29.94	78.72	62.78	34.19
–	GPT-Image-1.5	22.0	17.6	21.0	29.34	43.18	85.47	74.19	44.97
–	Nano Banana	16.4	20.0	18.0	16.14	22.72	78.59	68.37	30.24
–	Nano Banana Pro	39.6	44.4	41.0	34.99	51.35	88.65	69.84	50.38
Open-source Generative Models									
–	BAGEL	0.4	2.8	2.0	6.63	2.15	53.76	<u>52.72</u>	13.85
–	Z-Image	2.4	0.0	2.0	7.35	2.36	55.70	50.42	14.49
–	Qwen-Image	2.4	1.6	2.0	7.20	0.87	58.18	**59.35**	14.98
Agentic Image Generation Methods									
Mind-Brush	Qwen-IE-2511	25.2	6.4	15.8	17.42	1.66	49.16	41.88	16.74
Gen-Searcher	Qwen-IE-2511	<u>27.6</u>	12.0	<u>19.8</u>	<u>23.92</u>	**14.21**	<u>65.86</u>	45.22	<u>26.35</u>
GenEvolve	Qwen-IE-2511	20.8	<u>16.0</u>	18.4	15.75	<u>13.65</u>	63.32	48.16	22.91
Ranger	Qwen-IE-2511	**34.0**	**22.4**	**28.2**	**26.20**	13.19	**68.70**	51.25	**27.75**

The difficulty is sharpest in Fashion, where no baseline exceeds 40. Proprietary generation systems do not escape it with their built-in search either, scoring at most 34.9 on generation. These results show that existing methods are not yet ready for image generation in the wild, where the target must be found on a crowded and evolving web.

Generation is bounded by reference quality. A strong generator cannot compensate for incorrect references. Unify-Agent passes the same references to three generators, and even Nano Banana Pro lifts its score only to 31.6. Conversely, the same Nano Banana Pro scores 26.1 with its own search but 58.9 with references from Ranger. Gains in generation also follow gains in search: Fashion, where Ranger improves search the most, is also where its generation gains are largest, 16.9 and 21.0 points with the two generators.

Ranger leads the way. With only 8B parameters, **Ranger outperforms harness frameworks built on GPT-6-Astra and Claude-Opus-5**, raising the search score from 51.5 to 67.2 and achieving the best generation score in every domain with both generators. Its references are good enough that the open-source Qwen-Image-Edit-2511 (46.8) nearly matches the best baseline running on Nano Banana Pro (47.6). Compared with Gen-Searcher and GenEvolve, which share the same backbone, Ranger leads by more than 20 search points, demonstrating the effectiveness of the proposed paradigm. Comparisons on trajectories, generation quality and additional case studies are presented in Appendices I to K.

4.3 Generalization Performance

Although trained only on news-driven data, **Ranger achieves the best overall score among agentic methods on both benchmarks** (Table 2). Notably, these are the benchmarks proposed with Mind-Brush and Gen-Searcher, respectively, and Ranger outperforms each on its own benchmark, with 28.2 vs. 15.8 on Mind-Bench and 27.75 vs. 26.35 on KnowGen.

On Mind-Bench, Ranger brings clear gains on knowledge categories that hinge on a specific event or character, such as Special Events and New Movie Characters. On KnowGen, it stands out in visual correctness, which checks whether the depicted entity matches the ground truth and thus depends directly on the references, and it also leads in faithfulness to the prompt. These results show that the retrieval ability of Ranger carries over to benchmarks beyond Savanna.

4.4 Further Analysis

We examine whether each phase of the proposed paradigm addresses the challenge it targets. We also validate the reliability of the date filtering method in Appendix G and include a human-model alignment evaluation in Appendix H.

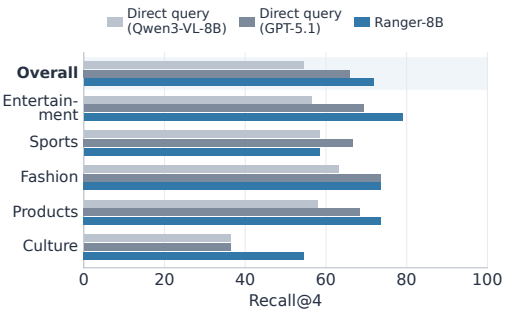


Figure 5: Comparison of Recall@4 of the first image search of direct query and Ranger.

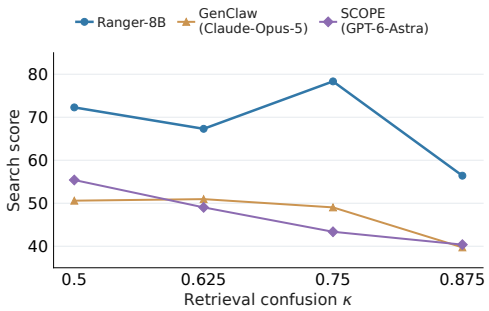


Figure 6: Search score grouped by retrieval confusion.

Locate recovers targets that direct search misses. We compare the first image search of Ranger with direct image searches directly derived from p by Qwen3-VL-8B and GPT-5.1 without text search. As shown in Fig. 5, Ranger reaches a Recall@4 of 71.7, compared with 54.4 and 66.0 for the two direct baselines. It proves that gathering textual evidence first makes the image query specific enough to find a newly introduced target, which addresses temporal novelty.

Corroborate resists semantic crowding. We group the 500 prompts by retrieval confusion κ (Eq. (2)) and compare Ranger with GenClaw (Claude-Opus-5) and SCOPE (GPT-6-Astra). As shown in Fig. 6, Ranger leads by at least 16 points in every group. In the most confusing group, it reaches a search score of 56.4, compared with 39.7 and 40.4 for the two baselines. Checking candidates across modalities and across images thus keeps the agent on the right target even when most search results show other entities, which addresses semantic crowding.

Isolate improves generation quality. On 102 prompts where Ranger calls `crop_image`, we compare the images generated from the cropped reference with those generated from the full source image. As shown in Fig. 7, cropping raises the identity score by 6.3 points with Qwen-Image-Edit-2511 and 3.8 points with Nano Banana Pro, while the context score improves less, by 1.4 and 1.3 points. These results show that removing the surrounding content helps the generator depict the right entity without losing the requested context, which addresses spatial entanglement. Generation examples are given in Appendix L.

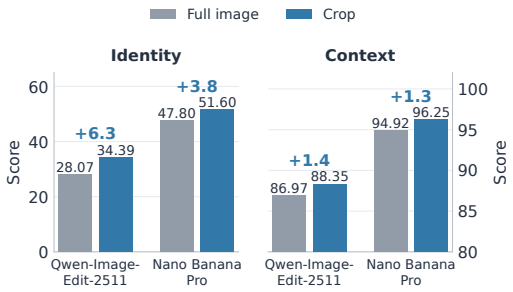


Figure 7: Generation scores with full and cropped references. The context axis starts at 80.

5 Conclusion

We studied agentic image generation in the wild and identified three challenges it poses: temporal novelty, semantic crowding, and spatial entanglement. We built a news-driven pipeline that reproduces these challenges under real search conditions, and used it to construct Savanna, on which even harness frameworks built on frontier models find a correct reference for only about half of the cases. We then proposed the Locate-Corroborate-Isolate paradigm. Trained on trajectories that follow it, Ranger-8B raises the search score from 51.5 to 67.2, gives the best generation quality with every generator we test, and surpasses prior agentic methods on other representative benchmarks. Our analysis further confirms that each phase addresses its intended challenge.

Ethics Statement

Our work complies with the ICLR Code of Ethics. Savanna and Ranger-15K are built entirely from publicly available news articles and web images, without collecting any private user data. The prompts focus on publicly released items such as products, outfits, vehicles, and artworks, and human involvement is limited to annotators verifying each benchmark sample. Because Ranger corroborates every reference against multiple sources before generation, it reduces the risk of depicting real-world entities incorrectly. The released benchmark, data, model, and code are intended for research purposes, and we discourage any use that produces misleading content.

Reproducibility Statement

We have taken several measures to ensure the reproducibility of our results. The paper details the data synthesis pipeline (Section 2.2), the meta queries used to collect news (Appendix E), the curation of Savanna (Section 2.3), the agent environment and tools (Section 3.1), and the training data and hyperparameters of Ranger-8B (Section 3.2). The evaluation metrics and full judge prompts are provided in Appendix F. We will release Savanna, Ranger-15K, the Ranger-8B checkpoint, and the code for data synthesis, training, and evaluation, so that others can reproduce our results and build new versions of Savanna from later news.

References

Anthropic. Introducing Claude Opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026a.

Anthropic. Introducing Claude Opus 5. https://www.anthropic.com/news/claude-opus-5, 2026b.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Shuang Chen, Quanxin Shou, Hangting Chen, Yucheng Zhou, Kaituo Feng, Wenbo Hu, Yi-Fan Zhang, Yunlong Lin, Wenxuan Huang, Mingyang Song, Dasen Dai, Bolin Jiang, Manyuan Zhang, Shi-Xue Zhang, Zhengkai Jiang, Lucas Wang, Zhao Zhong, Yu Cheng, and Nanyun Peng. Unify-agent: A unified multimodal agent for world-grounded image synthesis. *CoRR*, abs/2603.29620, 2026a.

Sixiang Chen, Zhaohu Xing, Tian Ye, Xinyu Geng, Yunlong Lin, Jianyu Lai, Xuanhua He, Fuxiang Zhai, Jialin Gao, and Lei Zhu. Genevolve: Self-evolving image generation agents via tool-orchestrated visual experience distillation. *CoRR*, abs/2605.21605, 2026b.

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. *arXiv preprint arXiv:2505.14683*, 2025.

Yifan Du, Zikang Liu, Jinbiao Peng, Jie Wu, Junyi Li, Jinyang Li, Wayne Xin Zhao, and Ji-Rong Wen. Towards long-horizon agentic multimodal search. *CoRR*, abs/2604.12890, 2026.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 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. *CoRR*, abs/2603.28767, 2026.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webwatcher: Breaking new frontier of vision-language deep research agent. *CoRR*, abs/2508.05748, 2025.

Google. Introducing Gemini 2.5 Flash Image, our state-of-the-art image model. https://developers.googleblog.com/introducing-gemini-2-5-flash-image/, 2025a.

Google. Introducing nano banana pro. https://blog.google/innovation-and-ai/products/nano-banana-pro/, 2025b.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026a.

Google. Nano Banana 2: Combining Pro capabilities with lightning-fast speed. https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/, 2026b.

Jun He, Junyan Ye, Zilong Huang, Dongzhi Jiang, Chenjue Zhang, Leqi Zhu, Renrui Zhang, Xiang Zhang, and Weijia Li. Mind-brush: Integrating agentic cognitive search and reasoning into image generation. *CoRR*, abs/2602.01756, 2026.

Chuanhao Li, Jianwen Sun, Yukang Feng, Mingliang Zhai, Yifan Chang, and Kaipeng Zhang. IA-T2I: internet-augmented text-to-image generation. *CoRR*, abs/2505.15779, 2025a.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. In *NeurIPS*, 2025b.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021.

Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *CoRR*, abs/2503.07265, 2025.

OpenAI. GPT-5.1: A smarter, more conversational ChatGPT. https://openai.com/index/gpt-5-1/, 2025a.

OpenAI. Introducing our latest image generation model in the API. https://openai.com/index/image-generation-api/, 2025b.

OpenAI. The new ChatGPT images is here. https://openai.com/index/new-chatgpt-images-is-here/, 2025c.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026a.

OpenAI. GPT-6 Astra: A new generation of intelligence. https://openai.com/index/gpt-6-astra/, 2026b.

OpenAI. Introducing ChatGPT images 2.0. https://openai.com/index/introducing-chatgpt-images-2-0/, 2026c.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4172–4182. IEEE, 2023.

Tianfei Ren, Zhipeng Yan, Yiming Zhao, Zhen Fang, Yu Zeng, Guohui Zhang, Hang Xu, Xiaoxiao Ma, Shiting Huang, Ke Xu, Wenxuan Huang, Lionel Z. Wang, Lin Chen, Zehui Chen, Jie Huang, and Feng Zhao. SCOPE: structured decomposition and conditional skill orchestration for complex image generation. *CoRR*, abs/2605.08043, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685. IEEE, 2022.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven C. H. Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Junhan Shi, Qilong Wu, Feng Yu, Chi Zhang, Shifeng Zhang, and Shilin Zhou. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *CoRR*, abs/2511.22699, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *CoRR*, abs/2504.12516, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming 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, and Zenan Liu. Qwen-image technical report. *CoRR*, abs/2508.02324, 2025.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search. In *ACL (1)*, pp. 2456–2487. Association for Computational Linguistics, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*. OpenReview.net, 2023.

Junyan Ye, Jun He, Zilong Huang, Dongzhi Jiang, Xuan Yang, Rui Chen, and Weijia Li. Genclaw: Code-driven agentic image generation. *CoRR*, abs/2605.30248, 2026.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. *CoRR*, abs/2505.14362, 2025.

A LLM USAGE STATEMENT

We used large language models (LLMs) to polish the writing of this paper. We also utilize LLMs to synthesize prompts (Section 2.2), serve as the teacher that produces training trajectories (Section 3.2), and judge search and generation results (Appendix F). All synthesized data pass the verification and filtering steps described in the paper, and every sample in Savanna is further reviewed by human annotators. We also used image generation tools to draw some of the logos in our figures. LLMs were not used to develop research ideas or to analyze experimental results. The authors have checked all content and take full responsibility for it.

B RELATED WORK

Generative models. Diffusion and unified multimodal models (Rombach et al., 2022; Peebles & Xie, 2023; Esser et al., 2024; Deng et al., 2025; Team et al., 2025) now produce high-quality images, and recent generators (Wu et al., 2025; OpenAI, 2025b;c; Google, 2025a;b) also accept reference images as conditions. Their knowledge, however, is bounded by training data, so they fail on uncommon entities (Niu et al., 2025). Conditioning on references of the entity fills this gap, which leaves open where the references come from. Subject-driven methods assume the user provides them, and retrieval-augmented methods draw them from a fixed database that cannot cover entities introduced after it was built. We instead obtain references from the live web and keep the generator fixed.

Multimodal search agents. Search agents began with language models that interleave reasoning with web search (Nakano et al., 2021; Yao et al., 2023), and were later trained with reinforcement learning to search over many turns (Li et al., 2025b), often on synthesized questions whose answers hide behind obscured clues (Wei et al., 2025). Multimodal agents then added image search (Wu et al., 2026; Geng et al., 2025; Du et al., 2026), and vision-language models learned to crop image regions for closer inspection (Zheng et al., 2025). All of these agents end with a text answer, so a retrieved image only needs to be understood. An agent for generation must hand the image itself to the generator, so the image has to show the right entity and little else. Ranger therefore crops to prepare the generator's input rather than to inspect, and our query synthesis draws difficulty from real search confusion rather than from obscured clues.

Agentic image generation. Agentic image generation places an agent before the generator. Early agents only reason about and rewrite the prompt, so they remain limited to what the model knows. Later agents add web search to fetch text and images, either as training-free harnesses around frontier models (Li et al., 2025a; He et al., 2026; Ren et al., 2026; Ye et al., 2026) or as trained agents (Feng et al., 2026; Chen et al., 2026b;a). Their benchmarks (Niu et al., 2025; He et al., 2026; Feng et al., 2026) mostly ask for entities with an established visual record, where the difficulty lies in knowing what to search for. For newly introduced entities, the harder part comes after search, namely telling the target apart from similar entities and isolating it into a clean reference. Savanna is built to test these two steps, and Ranger handles them in its Corroborate and Isolate phases.

C LIMITATIONS

Ranger changes only the inputs of a fixed downstream generator, so the final image still depends on how faithfully the generator follows the prompt and the reference images. We chose this setup on purpose, because it lets Ranger work with any generator that accepts reference images and lets it benefit directly as generators improve. Savanna is built from news published in July 2026, so it reflects the web at that time. We plan to release new versions built from more recent news at regular intervals (e.g., every six months) to keep the benchmark up to date. Finally, like other image generation systems, Ranger could be used to create misleading images of real people or products, so we plan to release it under a license that forbids such uses and to recommend that deployments keep the generator's safety filters enabled and watermark the generated images.

D TOOL IMPLEMENTATIONS

Table 3 summarizes the tools used in our agent environment and the news retrieval service used for data construction. Serper provides the search backends; the agent receives retrieved text or image pixels through the corresponding tool interface.

News dates and retrieval time windows. We collect news day by day through Serper's `/news` endpoint, setting both `cd_min` and `cd_max` in the `tbs` parameter to the queried day, which we record as the news date τ. For subsequent text and image searches, we set `tbs=cdr:1` with `cd_min` set to January 1, 2000 and `cd_max` set to $\tau + 3$ days. When returned results include timestamps, such as `date` or `datePublished`, we also parse these fields and remove results with unambiguous dates later than the cutoff. Results without a usable, unambiguous date are retained.

Table 3: Tool implementations. News retrieval is used during data construction; the remaining rows describe tools available to the agent.

Tool	Implementation	Inputs and returned evidence
News retrieval	Serper `/news`	Queries a meta query within a single-day date window and returns news titles, snippets, source URLs, and publisher names for data construction.
`text_search`	Serper `/search`	Accepts one to three queries and returns ranked organic results with titles, snippets, and URLs; the default is five results per query.
`image_search`	Serper `/images`	Searches by a text query, downloads candidate images, and displays up to five images with image IDs, titles, source-page URLs, and image URLs.
`web_fetch`	Internal tool	Retrieves webpage content from a URL according to a supplied prompt.
`crop_image`	Local image operation	Takes a retrieved image ID and a bounding box in normalized $[0, 1000]$ coordinates, then returns the cropped image with a new ID.

E META QUERIES

Table 4 lists all 50 meta queries used to collect news for Savanna. Each meta query names a family of visual artifacts within its subcategory, without pointing to a specific answer.

Table 4: Meta queries for each domain and subcategory of Savanna.

Domain	Subcategory	Meta queries
Entertainment	Film & TV	movie costume first look; television series costume debut; film set design first look; movie poster artwork unveiling
	Anime	anime key visual release; anime character design reveal
	Games	video game character costume reveal; video game vehicle model debut; video game expansion content announcement
	Collectibles	collectible figure reveal; comic issue variant cover reveal; collector edition box set reveal
	Music	album cover artwork reveal; concert stage design debut; music video costume design
Sports	General Sports	team kit design unveiling; championship trophy presentation; championship medal ceremony; player signing jersey presentation
	Motorsport	race car livery unveiling; motorsport driver team announcement; grand prix podium trophy; motorsport driver equipment debut
Fashion	Celebrities	celebrity press event appearance; red carpet gala arrival; celebrity street style outing
	Collections	fashion week runway collection; luxury accessory collection launch
	Watches & Jewelry	watch model product unveiling; high jewelry collection unveiling
Products	Tech	smartphone model product launch; wearable device product announcement; laptop computer product launch; gaming hardware product unveiling
	Automobiles	production car model debut; concept vehicle design unveiling; prototype camouflage road testing; vehicle cabin interior reveal
	Aviation & Space	airline aircraft livery unveiling; commercial aircraft cabin redesign; spacecraft mission hardware rollout; rocket engine test launch
	Consumer Goods	food beverage packaging redesign; cosmetics packaging redesign launch; household brand packaging redesign; brand logo identity unveiling
Culture	Art	gallery exhibition artwork debut; public art installation unveiling
	Architecture	cultural building design unveiling; new stadium design plans

F EVALUATION DETAILS AND JUDGE PROMPTS

Search score. The VLM compares the agent's references with the ground-truth image and returns a binary verdict for each query. A pass requires a usable view of the correct target, with identifying details clear enough to guide generation. Differences in viewpoint or background are allowed; a wrong version or an unusable view fails. Failed rollouts and missing references count as failures; judge errors are excluded from the denominator.

Generation score. The per-prompt score is

$$100\,(0.8\,s_{\mathrm{id}} + 0.2\,s_{\mathrm{ctx}}). \tag{3}$$

The identity score s_{id} checks whether e is drawn correctly: the VLM compares I_{gen} with the ground-truth reference and assigns 1, 0.2, or 0 for consistent, partially consistent, or inconsistent identity. When a prompt contains multiple targets, each target is scored separately and s_{id} is their average. The context score s_{ctx} is the fraction of rubrics, covering the scene, relations, and other context in p, that the VLM judges as satisfied. The prompt below defines the identity tiers and context judgments.

F.1 SEARCH JUDGE PROMPT

The judge receives the original user prompt, followed by the ground-truth image and the agent's reference images, with the following system prompt.

Table 5: System prompt for the search judge.

Search judge System message

You are grading a retrieval step inside an image-generation pipeline. An agent was given a prompt, searched the web, and returned a set of REFERENCE IMAGES that will be handed to an image-generation model together with that prompt.

You also get the GROUND-TRUTH image: the picture that provably shows the correct appearance of the target named by the prompt.

ONE QUESTION: if the image model receives the agent's images instead of the ground-truth image, will it draw the target correctly?

JUDGE FROM THE POINT OF VIEW OF DRAWING, NOT PIXEL SIMILARITY.

- PASS is correct when the agent's images show the SAME thing as the ground truth, from any angle, distance, crop, lighting or background, and clearly enough that its identifying details could be reproduced. A press photo, a product shot, an official render and a leaked picture of the very same design are interchangeable here.
- PASS is still correct when the agent's images add extra content, or show the target as part of a larger scene, as long as the target itself is visible and legible.
- FAIL when the target in the agent's images is a DIFFERENT thing: another edition, year, model, trim, colourway, person or named item. Near-identical siblings from the same series are the most important failures to catch — they look right and steer the generation wrong.
- FAIL when the agent returned no usable view of the target: only logos, crowds, venues, packaging, web-page screenshots, unrelated stock imagery, or a view too small, dark or obstructed for its details to be read.
- FAIL when the target is absent even though the images are otherwise about the right event.

Extra images that are irrelevant do NOT cause a failure by themselves, as long as at least one image lets the target be drawn correctly. But if the set as a whole would most likely lead the model to draw the wrong version, that is a FAIL.

IMAGE ORDER: image 0 is the GROUND TRUTH. The remaining images, in order, are the agent's reference images.

Return JSON only:

```
{"target": "<2-8 words: the thing the prompt actually asks to be
    drawn>",
 "gt_shows": "<2-8 words: what the ground-truth image shows of that
    target>",
 "agent_shows": "<2-8 words: what the agent's images show of that
    target>",
 "pass": true|false,
 "why": "<one short sentence: the concrete evidence -- the named
    difference if you fail it>"}
```

F.2 GENERATION JUDGE PROMPT

The judge receives the target description, original user prompt, context rubrics, generated image, and ground-truth images. Each image is explicitly marked as generated or ground truth. The system prompt is reproduced below.

Table 6: System prompt for the generation judge.

Generation judge — System message

You are an objective AI image evaluator. For one GENERATED IMAGE you produce two things: a TARGET CONSISTENCY TIER, and true/false verdicts on a list of CONTEXT criteria.

You receive:

1. The generated image. This is the ONLY image being scored.
2. One or more ground-truth photographs of the real target. They show what the target actually looks like. They are never the image being scored.
3. The name of the target and the original prompt.
4. CONTEXT criteria about the requested action, placement, setting, surrounding objects, counts, or spatial relationships.

Part 1: target consistency tier

The question is exactly this: is the target shown in the generated image THE SAME THING as the target shown in the ground-truth photographs?

It is NOT ”do the two images look alike”. The prompt deliberately asks for a different scene from the ground-truth photographs: a different moment, pose, camera angle, background, lighting and framing. That is by design. Those differences NEVER affect the tier.

Judge only identity-bearing features: text and its spelling, logo and emblem shape and where it sits on the target, colour scheme and how the colours are distributed, model or variant silhouette, patterns and textures, materials, distinctive structural parts.

Procedure:

a. Read the ground-truth photographs and list the target’s identity-bearing features. List only features actually visible in the ground-truth photographs.

b. For each listed feature, check the generated image and sort it into matched or mismatched.

c. Assign the tier.

Tier 1.0, CONSISTENT.
Every salient identity-bearing feature agrees. Someone familiar with this target would recognise it as this exact one. Rendering quality loss, stylisation and softened minor detail are acceptable: small text that is blurry but in the right place and the right colour still counts as matched. The question is ”is this the same entity”, not ”is this pixel-identical”.

Tier 0.2, PARTIALLY CONSISTENT.
The generated image is clearly attempting this target and gets at least one SPECIFIC identity feature right (the main colour scheme, the overall silhouette, the position of one marking), but at least one hard error makes it impossible to accept as the same entity. Hard errors: text misspelled or replaced by a different word, a logo replaced by a different logo or with a clearly different shape, the dominant colour scheme wrong, a clearly different model or variant.

Tier 0.0, INCONSISTENT.
The target is absent, or a different entity is depicted, or nothing beyond generic category membership agrees.

Continued on next page

Table 6 (continued)

Generation judge — System message

Hard rules for the tier. Apply them literally:

- Matching only the generic category is 0.0, NOT 0.2. "It is a cycling jersey", "it is a race car", "it is a necklace" earns nothing on its own. Tier 0.2 requires at least one specific identity feature to agree.
- Scene, background, pose, camera angle, lighting, framing and crop differences never change the tier.
- A feature not visible in the ground-truth photographs is outside the judgement. Do not list it and do not penalise it.
- If the generated image crops the target but every visible part agrees, judge the visible part; tier 1.0 stays available.
- If the target appears more than once in the generated image, judge the clearest instance.

Emit one entry per distinct target entity named by the target description. When only one entity is named, emit exactly one entry.

Part 2: context criteria

Judge each context criterion solely from the generated image. Ignore all content in the ground-truth photographs, including their background. Enforce the requested scene, action, placement, counts, other objects and spatial relations. Missing, wrong, illegible or unverifiable is false.

Output

Return a JSON object with exactly these keys:

```
{
  "entities": [{
    "name": "...",
    "gt_features": ["..."],
    "matched": ["..."],
    "mismatched": ["..."],
    "tier": 1.0,
    "reason": "at most 20 words"
  }],
  "context_results": [true, false],
  "context_reasons": ["short evidence", "short evidence"]
}
```

`tier` must be exactly 1.0, 0.2 or 0.0. `context_results` and `context_reasons` must each have exactly as many entries as the context criteria list, in the same order.

F.3 Introduction to Mind-Bench and KnowGen

Mind-Bench. Mind-Bench (He et al., 2026) contains 500 prompts across ten equally sized categories, divided into Knowledge and Reasoning. A VLM checks each generated image against a sample-specific checklist; a sample passes only if every item is satisfied. We report pass rates for both groups and overall.

KnowGen. KnowGen (Feng et al., 2026) contains 630 prompts in Science & Knowledge and Pop Culture & News. A VLM evaluates visual correctness, text accuracy, prompt faithfulness, and aesthetics on a $\{0, 0.5, 1\}$ scale. K-Score weights these dimensions by 0.4, 0.4, 0.1, and 0.1, respectively, and averages the two groups equally. We report all scores on a 0–100 scale.

G Prospective Validation of the Three-Day Retrieval Cutoff

In Section 2.2, we treat the news date as the issue time τ and restrict subsequent text and image searches to results published no later than $\tau + 3$ days. This date filtering approximates the web environment shortly after an event, when newly introduced entities exhibit temporal novelty. We assess this approximation by comparing image-search results collected in real time with those retrieved later under a three-day date cutoff.

We prospectively tracked 50 entities extracted using our pipeline to measure changes in visual-reference retrieval during the first week after release. For each entity, we verified its release time τ from its initial public announcement and fixed an image-search query at enrollment. We collected the top eight results within 24 hours of release and again on days 3 and 7, without date filters. Each target was queried twice in each condition. We report Hit@8, the percentage of queries with at least one correct visual reference among the top eight results, averaged over the two repetitions.

To assess retrospective date filtering, we also queried the same targets on day 7 with the same fixed queries, restricting results to those published no later than $\tau + 3$ days. We queried each target twice in this condition and compared the results with those collected on day 3 without date filters.

Table 7: Visual-reference retrieval for 50 prospectively tracked entities. Conditions A–C use searches without date filters at the indicated times; condition D searches on day 7 for results published no later than day 3. Release time τ is verified from each entity's initial public announcement.

Condition	Actual retrieval time	Publication-date cutoff	Hit@8 (%) ↑
A: Initial release	Within 24 hours of τ	None	68.0
B: Day 3	$\tau + 3$	None	76.0
C: Day 7	$\tau + 7$	None	80.0
D: Retrospective cutoff	$\tau + 7$	$\tau + 3$	78.0

As shown in Table 7, Hit@8 increased from 68.0% within 24 hours of release to 76.0% on day 3 and 80.0% on day 7. Correct visual references thus became easier to retrieve over the first week in this sample. Searching on day 7 with the three-day publication-date cutoff yielded 78.0%, close to the 76.0% observed on day 3. This comparison supports the use of date filtering to approximate visual-reference retrieval shortly after an event, consistent with the temporal-novelty setting used in our synthesis pipeline.

H Human-Model Alignment

We validate the VLM evaluator against human judgments using 100 images selected from Savanna. Human annotators assess the search score and the identity and context components of the generation score, following the criteria in Appendix F. Search judgments check whether retrieved references match the ground-truth target; generation judgments assess target identity in the generated image and satisfaction of the context rubrics.

Table 8: Agreement between human annotators and the VLM evaluator in the 100-image validation study on Savanna. Identity and context are the two components of the generation score.

Metric	Judgment	Agreement (%)
Search score	Reference matches ground truth	98.0
Generation score: identity	Target identity matches ground truth	98.0
Generation score: context	Context rubric satisfaction	94.5

As shown in Table 8, the VLM agrees with human annotators on 98% of search judgments and 98% of generation identity judgments. Agreement on context rubrics is slightly lower at 94.5%, but remains high despite the finer-grained criteria. These results support the use of a strong teacher VLM to judge retrieval correctness against ground-truth references and to evaluate target identity and contextual details on Savanna.

I Comparison of Agent Trajectories



Figure 8: Pre-generation trajectories for the same target. Excerpts are verbatim; tool names are normalized and verdicts annotated. Green/amber/red mark grounded decisions, uncertainty, and incorrect or missing references. These independent runs omit SCOPE's later verification and repair. Ground truth is shown for comparison only.

J Qualitative Generation Comparison on Savanna

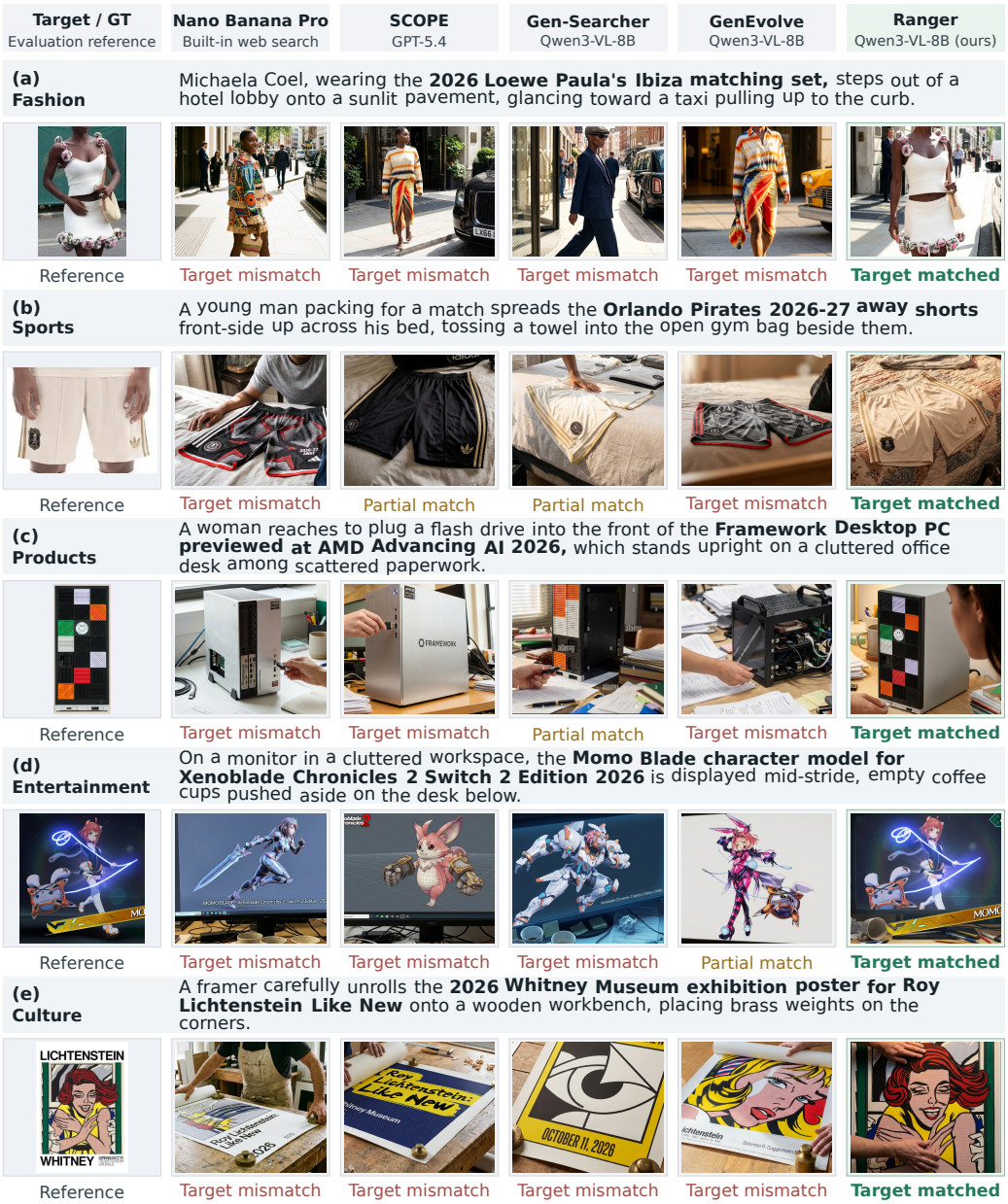


Figure 9: Generation comparisons across all five Savanna domains, using Nano Banana Pro with built-in search or agent-selected references (SCOPE: GPT-5.4). Ground truth is for evaluation only. Labels assess target identity on original outputs; display crops emphasize the target.

K GALLERY OF GENERATED IMAGES

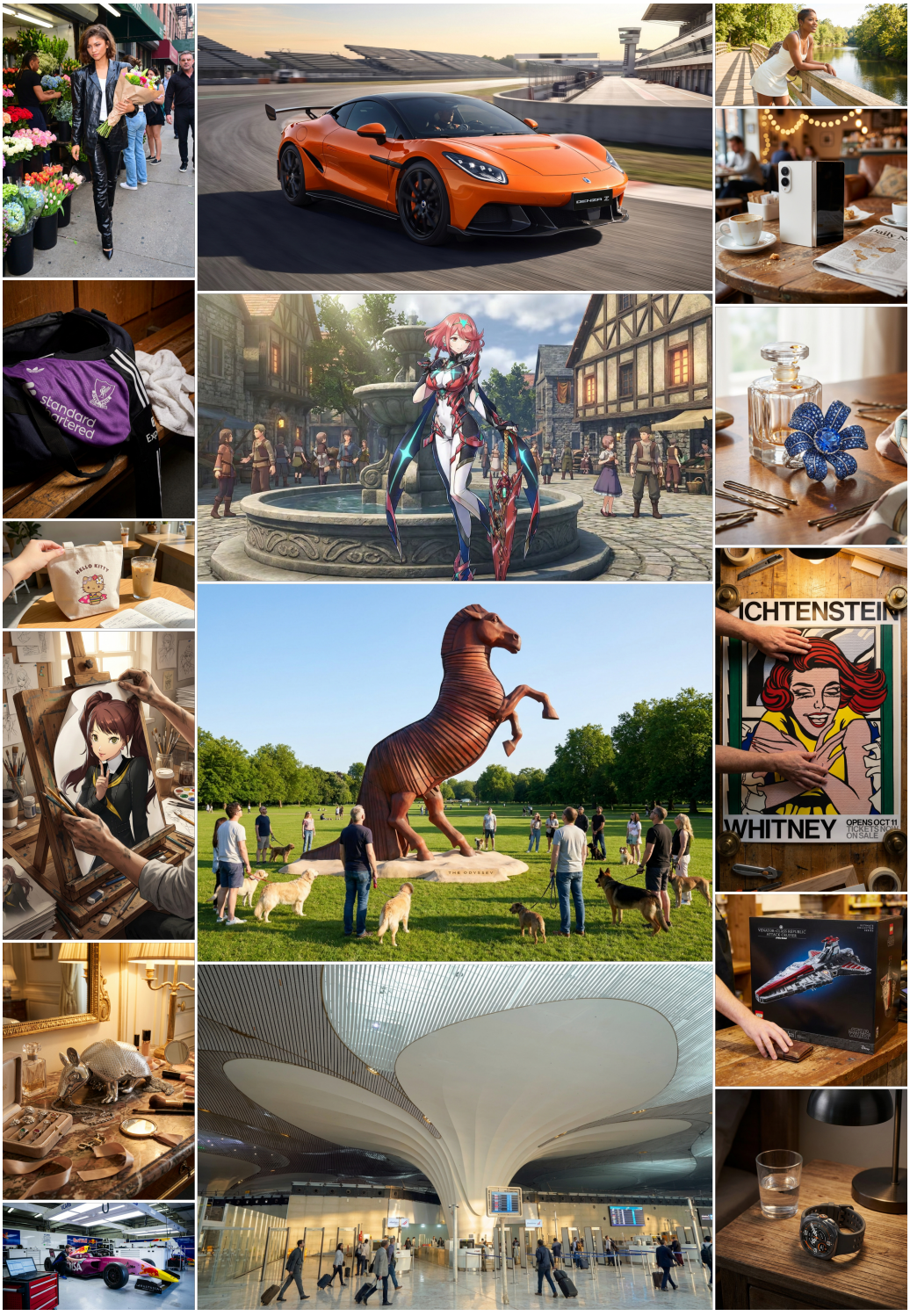
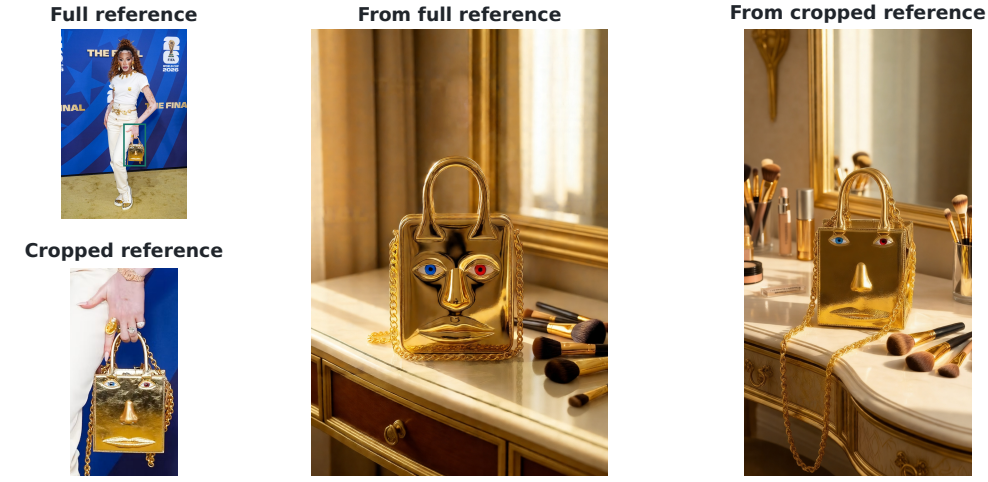


Figure 10: Sixteen Nano Banana Pro outputs from Ranger prompts and references, spanning all five Savanna domains. Images are shown uncropped at their original aspect ratios.

L Effect of Reference Cropping

(a) Qwen-Image-Edit-2511: Schiaparelli face bag

Sitting upright on a hotel dressing table beside scattered makeup brushes, the Schiaparelli face bag Winnie Harlow carried at the 2026 World Cup catches the morning light.



(b) Nano Banana Pro: T-bar shoes

A man steps onto a tiled foyer floor wearing the shoes from Richard E. Grant's Wimbledon 2026 outfit, setting a canvas grocery bag down beside his ankles.



Figure 11: Reference-cropping comparisons with Qwen-Image-Edit-2511 and Nano Banana Pro. Each case shows the user prompt, full and cropped references, and paired generations. Green boxes mark the reference crops.

With the full reference, Qwen-Image-Edit adds prominent eyebrows and oversized facial features to the bag; the crop yields a flatter face with smaller eyes. For Nano Banana Pro, the crop restores the shoes' central T-bar and horizontal straps, replacing the isolated side cutouts in the full-reference output.