RAVEL: Rare Concept Generation and Editing via Graph-driven Relational Guidance

Kavana Venkatesh[1] **Yusuf Dalva**[1] **Ismini Lourentzou**[2] **Pinar Yanardag**[1]
[1]Virginia Tech [2]UIUC
https://ravel-diffusion.github.io/

Abstract

Text-to-image (T2I) models often omit or misrepresent the defining attributes of rare and culturally specific concepts. We introduce **RAVEL**, a training-free framework that improves rare concept generation across diffusion and autoregressive image generators through knowledge graph-based retrieval-augmented generation. RAVEL retrieves visual, relational, and contextual attributes from a knowledge graph constructed from textual sources and uses a large language model to incorporate them into text conditioning. This conditioning emphasizes distinguishing attributes and relevant relationships without requiring reference images for T2I generation. To further improve RAVEL's attribute accuracy, we introduce *Self-Correcting RAG-Guided Diffusion (SRD)*, an optional module that uses a vision-language model to check generated images against the retrieved attributes and refine the conditioning for subsequent generations. Beyond guiding generation and refinement, the retrieved attributes and relationships also inform editing instructions, extending RAVEL to context-aware image editing. For evaluation, we introduce three benchmarks, *MythoBench*, *Rare-Concept-1K*, and *NovelBench*, covering cultural symbolism, fine-grained concept attributes, and literary descriptions. Experiments with Stable Diffusion XL, FLUX.1 [dev], DALL-E 3, GLM-Image, and Janus-Pro show that RAVEL achieves substantial improvements in attribute accuracy and text–image alignment over the corresponding base generators and outperforms the evaluated retrieval and prompt-enhancement baselines. Human preference evaluations further support these improvements over the corresponding base generators.

1 Introduction

Text-to-image (T2I) models have made substantial progress in generating visually rich images from natural language, with advances in both diffusion [12, 27] and autoregressive [5] generation. However, accurately depicting rare and culturally specific concepts remains challenging [29, 4]. An image may appear plausible while missing the distinctive morphology, symbolic objects, or relationships that identify the intended concept. As illustrated in Fig. 1, the base generator can produce images that do not faithfully depict the requested concepts. These failures motivate making relevant concept knowledge explicit in the conditioning supplied to the image generator.

Several strategies can provide additional information to condition the generation process. Manual prompt engineering makes desired content more explicit [34, 19], but requires users to identify and describe the relevant details. Personalization methods such as textual inversion [9] learn concept-specific representations from reference images through additional optimization. Image-based retrieval-augmented generation (RAG) methods supply relevant visual examples at inference [4, 31, 22]; ImageRAG also uses feedback on generated images to determine what to retrieve [31]. These approaches demonstrate the value of external knowledge. We investigate a complementary setting in

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: **Concept generation with RAVEL.** Comparisons between FLUX and FLUX + RAVEL illustrate improvements in identifying the requested concepts and depicting their defining visual attributes. RAVEL incorporates attributes and relationships retrieved from a knowledge graph into text conditioning. Reference images are shown only for comparison and are not provided during generation.

which textual sources provide the defining attributes and relationships needed for generation via a knowledge graph structure, without requiring reference images at inference.

We introduce **RAVEL**, a training-free framework for rare concept generation across diffusion and autoregressive image generators. RAVEL organizes information from textual sources in a directed, multi-relational knowledge graph (KG), representing visual, relational, and contextual attributes together with typed relationships between entities. Given a user request, our framework retrieves attributes of the target entities and relevant relationships to connected entities. A large language model (LLM) incorporates this information into text conditioning that emphasizes distinguishing features and contextual associations. The graph makes these associations explicit and allows related information to be retrieved together. RAVEL uses the image generator's existing text interface and requires no additional model fine-tuning.

To further improve RAVEL's attribute accuracy, we introduce *Self-Correcting RAG-Guided Diffusion (SRD)*, an optional module that uses a vision-language model (VLM) to check generated images against the retrieved attributes. The resulting feedback guides conditioning refinement for subsequent generations, reusing the information retrieved for the initial image. Beyond guiding generation and refinement, these attributes and relationships also inform editing instructions, extending RAVEL to context-aware image editing. We evaluate RAVEL on three benchmarks covering cultural concepts, fine-grained concept attributes, and literary descriptions. Experiments across five image generators show substantial improvements in attribute accuracy and text-image alignment over the corresponding base generators, along with gains over the evaluated retrieval and prompt-enhancement baselines. Human preference evaluations further support these improvements over the base generators included in the study.

2 Related Work

LLMs and RAG for Image Generation. Retrieval-augmented generation conditions language models on external documents [16], while KG-augmented language models retrieve relational facts for complex question answering [30]. In image generation, scene-graph methods guide synthesis with objects and their relationships [14], while LayoutGPT, LLM-grounded Diffusion, and Ranni use LLMs to construct layouts or semantic control panels [7, 18, 8]. LLM Blueprint adds iterative object refinement [10], and PCIG uses a knowledge graph to predict object locations [32]. Retrieval-Augmented Diffusion Models and Re-Imagen provide visual context from external image databases [2, 4]. ImageRAG selects reference images using VLM feedback without RAG-specific training [31], whereas RealRAG trains a retriever through self-reflective contrastive learning [22]. RAVEL retrieves concept attributes and typed relationships from a KG constructed from textual sources and uses an LLM to incorporate them into text conditioning. This supports diffusion and autoregressive image generators through their existing text interfaces, without reference images for T2I generation or additional model fine-tuning.

Image Editing and Self-Correction. DiffEdit [6] derives editing masks from text-conditioned diffusion predictions, and Prompt-to-Prompt [11] controls edits through cross-attention. InstructPix2Pix [3] learns from paired editing examples, while HIVE [40] incorporates human feedback into

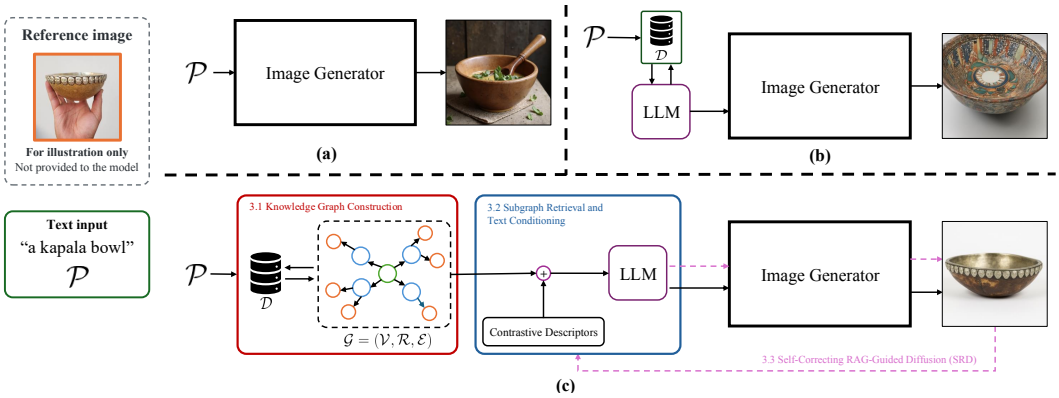

Figure 2: **Overview of RAVEL.** (a) **Direct T2I generation** produces a generic wooden bowl from the prompt $\mathcal{P}$. (b) **Unstructured retrieval** enriches conditioning using an external database $\mathcal{D}$ but produces a decorated bowl that differs from the requested concept. (c) **RAVEL** constructs a knowledge graph $\mathcal{G}$ from textual sources (red). An LLM combines retrieved attributes and relationships with contrastive descriptors to construct text conditioning (blue). The optional SRD loop (dashed) uses VLM-based attribute verification to refine conditioning for subsequent generations. The "kapala bowl" example illustrates improved depiction of skull motifs and metallic appearance. *The reference image is for illustration only and is not provided to the models.*

model fine-tuning. For iterative correction, SLD [35] uses an LLM controller to assess prompt alignment and apply corrective image operations, while ReflectionFlow [41] uses textual reflections and a reflection-tuned diffusion model. RAVEL's contribution is to supply retrieved concept knowledge for editing and verification: it enriches editing instructions in our ControlNet-based pipeline, while SRD checks generated images against retrieved attributes and refines text conditioning for subsequent generations. SRD reuses the initial retrieval and requires no additional model training.

3 Method

RAVEL is a training-free framework that improves semantic grounding in diffusion and autoregressive image generation for rare concepts. As illustrated in Fig. 2, it constructs a knowledge graph from textual sources, retrieves concept-specific attributes and relationships, and incorporates them into text conditioning. The optional SRD module then uses VLM feedback to check generated images against the retrieved visual attributes and refine the conditioning for subsequent generations. The retrieved knowledge also enriches editing instructions, extending RAVEL to context-aware image editing.

3.1 Knowledge Graph Construction

Given a collection of textual sources $\mathcal{D}$, we construct a directed, multi-relational knowledge graph $\mathcal{G} = \Phi(\mathcal{D})$, represented as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$. In this entity-level representation, $\mathcal{V}$ is the set of nodes representing entities, and $\mathcal{R}$ is the set of directed edges encoding typed semantic relationships between them. The set $\mathcal{E} = \{(e, \mathbf{a}_e) : e \in \mathcal{V}\}$ associates each node e with a structured attribute record $\mathbf{a}_e$.

Attribute extraction. For each entity e, an LLM extracts an attribute record $\mathbf{a}_e = f_{\text{ext}}(\mathcal{D}_e)$ from its source documents $\mathcal{D}_e \subseteq \mathcal{D}$, where $\mathbf{a}_e = (a_e^{\text{vis}}, a_e^{\text{func}}, a_e^{\text{rel}}, a_e^{\text{ctx}})$. These groups describe visual attributes (appearance and morphology), functional attributes (roles and behavior), relational attributes (associations with other entities), and contextual attributes (geographic, historical, cultural, or narrative information). We validate the resulting textual descriptions and relationships through source cross-referencing, consistency checks, and manual review before constructing the data. Appendix H.3 details this procedure and the knowledge-graph audit.

Entity relation construction. Let $\mathcal{L}$ denote the set of relationship types. We represent directed edges as triples $(u, r, v) \in \mathcal{R} \subseteq \mathcal{V} \times \mathcal{L} \times \mathcal{V}$, where u and v are entity nodes and r specifies the relationship type. To support traversal in both directions, we include a reciprocal edge for every

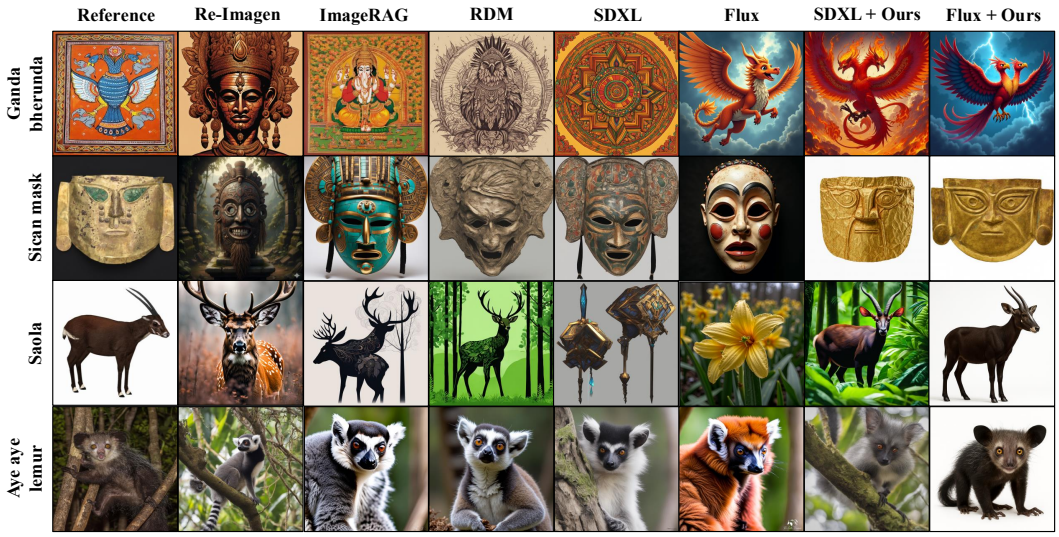

Figure 3: **Rare-concept generation against diffusion and retrieval baselines.** Comparisons on biological and cultural concepts illustrate differences in defining attributes. *Reference images are shown for visual comparison. RAVEL does not receive a reference image as input.*

triple:

$$\forall (u, r, v) \in \mathcal{R}, \qquad (v, r^{-1}, u) \in \mathcal{R}. \tag{1}$$

Here, r^{-1} denotes the inverse relationship type, with $(r^{-1})^{-1} = r$ and $r^{-1} = r$ for symmetric relationships. For example, `POLLINATES` and `POLLINATED_BY` form an inverse pair. Appendix H provides further details on the extraction schema and graph construction.

3.2 Subgraph Retrieval and Text Conditioning

Given a user prompt $\mathcal{P}$, RAVEL identifies the target entities $\mathcal{V}_{\text{sub}} \subseteq \mathcal{V}$, either named directly or identified through relationships specified in the prompt. Their attributes are retrieved along with those of a limited number of neighboring entities reachable within k hops by traversing edges in either direction. By default, $k = 1$, so only directly connected neighbors are considered.

RAVEL also retrieves a limited number of relationship triples $(u, r, v) \in \mathcal{R}$, where r describes the relationship from entity u to entity v. Each retrieved triple involves at least one target entity, so $u \in \mathcal{V}_{\text{sub}}$ or $v \in \mathcal{V}_{\text{sub}}$. We denote the retrieved attributes and relationships by $\mathcal{C}_{\mathcal{P}}$ and use them to construct text conditioning. The LLM uses the target entities' contrastive descriptors to emphasize features that distinguish them from similar concepts in the text conditioning. When SRD is enabled, it checks whether the visual attributes retrieved for those entities are present in the generated image.

Contrastive chain-of-thought (CoT). We use an LLM to construct text conditioning $\mathcal{P}' = f_{\text{syn}}(\mathcal{P}, \mathcal{C}_{\mathcal{P}})$ from the user request and the attributes and relationships retrieved from the graph $\mathcal{G}$. Entity attributes describe the target, relationship triples specify its associations, and stored contrastive descriptors clarify differences from similar concepts. For example, retrieved "upright, torch-like" attributes and a stored contrast with hanging forms can be expressed as "upright and torch-like, rather than hanging." The resulting text conditions the generator without updating model parameters, while the retrieved attributes, relationships, and descriptors are cached for optional SRD refinement (see Appendix H.4).

3.3 Self-Correcting RAG-Guided Diffusion (SRD)

To further improve attribute accuracy, SRD uses VLM-based feedback to refine text conditioning. Starting from $\mathcal{P}_0 = \mathcal{P}'$, the image generator produces $\mathcal{I}_0$. SRD then checks the image against the retrieved visual attributes and uses the feedback to guide subsequent generations.

Table 1: **Rare-concept generation.** Mean scores and standard deviations. Bold marks the best overall score; underlining marks the best unmodified generator. DINO and VQA denote DINOv2 and BLIP-VQA.

Method	CLIP-T ↑	CLIP-I ↑	DINO ↑	VQA ↑	LPIPS ↓	SigLIP ↑
Baseline Models						
FLUX [15]	0.187 ± .012	0.418 ± .025	0.287 ± .019	0.228 ± .018	0.580 ± .031	0.063 ± .007
SDXL [27]	0.184 ± .013	0.297 ± .021	0.262 ± .017	0.199 ± .016	0.623 ± .034	0.034 ± .005
DALL-E 3 [24]	0.220 ± .014	0.480 ± .027	0.367 ± .022	0.279 ± .020	0.568 ± .030	0.069 ± .008
GLM-Image [36]	0.232 ± .013	0.533 ± .026	0.391 ± .021	0.297 ± .019	0.612 ± .032	0.052 ± .007
Janus-Pro [5]	0.261 ± .014	0.571 ± .028	0.427 ± .023	0.331 ± .021	0.584 ± .030	0.058 ± .007
Retrieval and Rare-Concept Methods						
ImageRAG [31]	0.284 ± .026	0.518 ± .011	0.288 ± .002	0.216 ± .001	0.654 ± .030	0.042 ± .010
RDM [2]	0.299 ± .021	0.510 ± .010	0.302 ± .010	0.235 ± .013	0.591 ± .015	0.049 ± .012
ReCon [21]	0.279 ± .012	0.499 ± .031	0.281 ± .012	0.222 ± .013	0.668 ± .010	0.038 ± .020
SeedSelect [29]	0.198 ± .010	0.480 ± .021	0.279 ± .011	0.201 ± .032	0.671 ± .012	0.037 ± .020
LLM-Enhanced Methods						
LMD [18]	0.201 ± .012	0.433 ± .010	0.277 ± .045	0.211 ± .031	0.636 ± .015	0.037 ± .013
LLM-Blueprint [10]	0.190 ± .010	0.456 ± .013	0.271 ± .012	0.194 ± .011	0.651 ± .011	0.031 ± .010
Ranni [8]	0.190 ± .011	0.429 ± .011	0.262 ± .022	0.200 ± .012	0.657 ± .010	0.031 ± .021
Ours (RAVEL)						
SDXL + Ours	0.301 ± .016	0.686 ± .032	0.569 ± .027	0.412 ± .022	0.323 ± .020	0.075 ± .009
FLUX + Ours	0.331 ± .017	0.681 ± .033	0.582 ± .028	0.422 ± .024	0.288 ± .019	0.087 ± .010
DALL-E 3 + Ours	**0.336 ± .018**	**0.746 ± .035**	**0.642 ± .029**	**0.448 ± .026**	**0.266 ± .017**	**0.093 ± .010**
GLM-Image + Ours	0.305 ± .015	0.682 ± .011	0.571 ± .023	0.409 ± .009	0.412 ± .026	0.081 ± .010
Janus-Pro + Ours	0.332 ± .011	0.709 ± .022	0.598 ± .017	0.444 ± .028	0.398 ± .029	0.087 ± .013

Attribute Coverage Score (ACS). Let $\mathcal{A}_{\text{ret}} = \{a_1, \ldots, a_N\}$ denote the visual attributes retrieved from the primary entity records for verification, with $N > 0$. At round t, a VLM assigns $b_{t,i} = 1$ if it judges attribute a_i present in $\mathcal{I}_t$, and $b_{t,i} = 0$ otherwise. We define ACS as the fraction of these attributes judged present, $\text{ACS}_t = \frac{1}{N}\sum_{i=1}^{N} b_{t,i}$. ACS measures verified attribute coverage in the current image, using the same retrieved target set at every round.

Conditioning refinement. If $\text{ACS}_t < \tau$ and the refinement budget remains, SRD identifies the attributes judged missing, $\mathcal{A}_{\text{miss},t} = \{a_i \in \mathcal{A}_{\text{ret}} : b_{t,i} = 0\}$. It uses this feedback and the cached context to refine the conditioning and generate a new image:

$$\mathcal{P}_{t+1} = f_{\text{refine}}(\mathcal{P}_t, \mathcal{A}_{\text{miss},t}, \mathcal{C}_{\mathcal{P}}), \quad \mathcal{I}_{t+1} = \text{Generate}(\mathcal{P}_{t+1}).$$

As summarized in Algorithm 1, SRD returns the current image when $\text{ACS}_t \geq \tau$, with $\tau = 0.85$, or after $K_{\max} = 3$ refinement updates following the initial generation. Appendix E evaluates this refinement process by examining how often the threshold is reached within this budget and how well previously satisfied attributes are retained across rounds.

3.4 Context-Aware Image Editing

RAVEL also uses retrieved knowledge to enrich editing instructions for an input image $\mathcal{I}_{\text{in}}$. Given an instruction $\mathcal{P}_e$, it retrieves attributes and relationships relevant to the requested modification, reusing previously retrieved attributes and relationships when applicable or querying the graph for newly introduced entities. An LLM incorporates this information into the editing text $\mathcal{P}'_e$, which is supplied with the input image to the editing pipeline. Our experiments use a ControlNet-based pipeline to evaluate this conditioning mechanism. This editing procedure operates on an input image, while SRD regenerates images from refined text conditioning.

4 Experiments

We demonstrate that **RAVEL** improves rare-concept generation across five image generators and supports context-aware image editing and self-correction. Comparisons with retrieval-based and LLM/VLM baselines show consistent gains in attribute accuracy across diverse concept categories and generator architectures. We further quantify attribute recovery and the inference cost of iterative refinement.

Table 2: **Comparison with Baseline Models.** We report (Baseline/**+Ours**) performance across benchmarks. Δ_{avg} denotes average improvement across all backbones. Best result in each row is shown in bold.

Benchmark	Metric	SDXL	Flux	DALL-E3	GLM-Img	Janus Pro	Δ_{avg}
T2I-Comp-Bench [13]	Comp. Acc.	0.512 / 0.721	0.587 / 0.798	0.623 / 0.832	0.561 / 0.795	0.582 / **0.836**	+0.223
	CLIP-I	0.273 / 0.662	0.318 / 0.701	**0.351 / 0.728**	0.301 / 0.702	0.322 / 0.715	+0.389
	BLIP-VQA	0.212 / 0.424	0.251 / 0.453	**0.279 / 0.468**	0.238 / 0.441	0.249 / 0.452	+0.202
MythoBench	Attr. Acc.	0.347 / 0.779	0.407 / 0.865	0.500 / 0.882	0.392 / 0.828	**0.428 / 0.851**	+0.426
	Context Rel.	0.457 / 0.849	0.553 / 0.884	**0.560 / 0.924**	0.502 / 0.871	0.531 / 0.889	+0.363
	Intent Repr.	0.332 / 0.809	0.316 / 0.899	0.512 / 0.854	0.361 / 0.872	**0.402 / 0.902**	+0.483
	Visual Fid.	0.419 / 0.762	0.504 / 0.812	**0.539 / 0.829**	0.463 / 0.794	0.491 / 0.812	+0.319
Rare-1K	Attr. Acc.	0.398 / 0.741	0.467 / 0.812	0.523 / 0.845	0.441 / 0.812	**0.468 / 0.852**	+0.353
	Context Rel.	0.421 / 0.698	0.501 / 0.789	**0.548 / 0.821**	0.462 / 0.736	0.487 / 0.754	+0.276
	Intent Repr.	0.389 / 0.712	0.445 / 0.801	**0.537 / 0.834**	0.423 / 0.749	0.451 / 0.773	+0.325
	Visual Fid.	0.435 / 0.739	0.501 / 0.794	**0.534 / 0.818**	0.472 / 0.764	0.498 / 0.781	+0.291
NovelBench	Attr. Acc.	0.421 / 0.765	0.512 / 0.843	**0.567 / 0.878**	0.463 / 0.796	0.495 / 0.821	+0.329
	Context Rel.	0.448 / 0.812	0.534 / 0.871	**0.589 / 0.901**	0.487 / 0.839	0.516 / 0.859	+0.342
	Intent Repr.	0.412 / 0.789	0.478 / 0.856	0.543 / 0.867	0.446 / 0.845	**0.478 / 0.872**	+0.374
	Visual Fid.	0.432 / 0.758	0.507 / 0.826	**0.551 / 0.851**	0.468 / 0.785	0.498 / 0.804	+0.314

4.1 Experimental Setup

Models and baselines. We evaluate Stable Diffusion XL (SDXL) [27], FLUX.1 [dev] [15], DALL-E 3 [24], GLM-Image [36], and Janus-Pro [5], covering diffusion, autoregressive, and hybrid generators. Comparisons include the base generators, retrieval and rare-concept methods (ImageRAG, RDM, ReCon, and SeedSelect), and LLM-based approaches (LMD, LLM-Blueprint, and Ranni), listed in Table 1, as well as LLM/VLM prompt enhancement. We compare editing with ControlNet [38] and self-correction with SLD [35] and ReflectionFlow [41].

Datasets. Our benchmark suite contains 2,000 prompts covering 600 rare concepts across more than 15 categories. MythoBench contains 500 character prompts organized into simple ($\leq$ 3 attributes), composite (4–6 attributes), and relational ($\geq$ 7 attributes with explicit dependencies) tiers. Rare-Concept-1K contains 1,000 prompts spanning obsolete technologies, rare creatures, cultural artifacts, architectural objects, and uncommon tools. NovelBench contains 500 prompts derived from public-domain literary descriptions, including Project Gutenberg. Together, these benchmarks cover morphological, cultural, and narrative attributes; their construction is detailed in Appendix F. Generation evaluation uses 500 prompts per benchmark, totaling 1,500 per backbone; editing and self-correction use 500 images spanning 38 characters.

Inference setup. RAVEL uses GPT-4o [1] to identify target entities and construct text conditioning from attributes and relationships retrieved from a knowledge graph stored in Neo4j [23]. The resulting conditioning is supplied to the selected image generator, which is either run locally on a single NVIDIA A100 or accessed through its provider API. For locally deployed diffusion backbones, we use guidance scales in $[5, 15]$.

Evaluation. GPT-5.2 [25] scores Attribute Accuracy, Context Relevance, Visual Fidelity, and Intent Representation using a G-Eval-based protocol [20]. Evaluation targets are prepared in a separate annotation pass, manually checked against source documentation, and held fixed across methods, although source overlap with the knowledge graph remains possible (Appendix F.2). We report CLIP-T [28] and SigLIP [37] for text–image alignment, CLIP-I [28], DINOv2 [26], and LPIPS [39] for reference-image similarity, and BLIP-VQA [17] for attribute evaluation. Image–image metrics use shared surrogate references for generation and the input image for editing; surrogate references are not provided to the generator (Appendix F.3). For each concept, we select a surrogate from 3–10 candidate images after retaining candidates that satisfy at least 80% of the documented defining attributes. Every method is evaluated against the same selected reference. These metrics complement the attribute-based judgments by measuring alignment and visual similarity from different representations. For editing, similarity to the input measures preservation and must be considered alongside the requested modification, since an unchanged image can have low LPIPS without completing the edit.

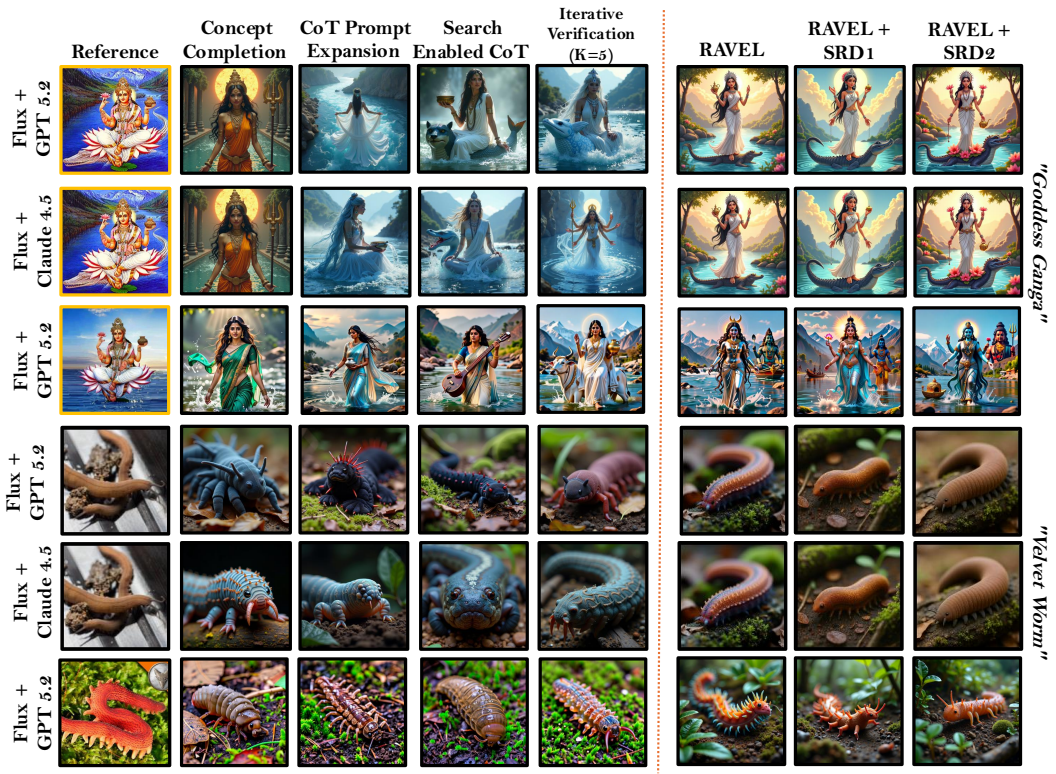

Figure 4: **Comparison with LLM/VLM prompt enhancement.** FLUX generations of *Goddess Ganga* and *Velvet Worm* using GPT-5.2 and Claude Haiku 4.5 for concept completion, CoT expansion, search augmentation, and iterative VQA verification ($K = 5$), compared with RAVEL and SRD. Reference images are not provided during generation.

4.2 Rare Concept Generation

Generation metrics. Table 1 shows that RAVEL improves all six metrics across diffusion, autoregressive, and hybrid generators. Gains over the corresponding base models and the compared retrieval and LLM-based methods span text–image alignment, attribute evaluation, and similarity to shared references.

Semantic and compositional alignment. Table 2 evaluates whether defining attributes are depicted in the intended context. RAVEL improves all evaluation dimensions across MythoBench, RareConcept-1K, and NovelBench; MythoBench shows strongest improvements in Attribute Accuracy and Intent Representation. T2I-CompBench further shows gains in compositional accuracy, CLIP-I, and BLIP-VQA for every evaluated backbone.

Qualitative comparisons. In Fig. 3, RAVEL more closely reproduces the two-headed form of *Ganda bherunda*, the characteristic designs of the *Sican mask*, and the morphology of *Saola* and *Aye-aye*. Reference images are shown for comparison only and are not supplied during generation. Appendix B provides further examples across biology, mythology, and literature.

Figure 4 compares RAVEL with GPT-5.2 and Claude Haiku 4.5 using concept completion, CoT expansion, search augmentation, and iterative VQA refinement with up to five updates. Each comparison uses the same generator and normalized prompt formatting. The illustrated baselines retain errors in *Goddess Ganga*'s symbolic composition and *Velvet Worm*'s morphology; RAVEL improves these attributes, with SRD correcting remaining errors. Appendix C.3 reports the quantitative comparisons.

Figure 5 extends the evaluation to hybrid GLM-Image and autoregressive Janus-Pro. RAVEL better reproduces the fine filaments of *Hair Ice*, spherical morphology of *Marimo*, combination of animal features in of *Fei Lian*, and appearance of the literary character of *Uncle Wiggily*. The GLM-

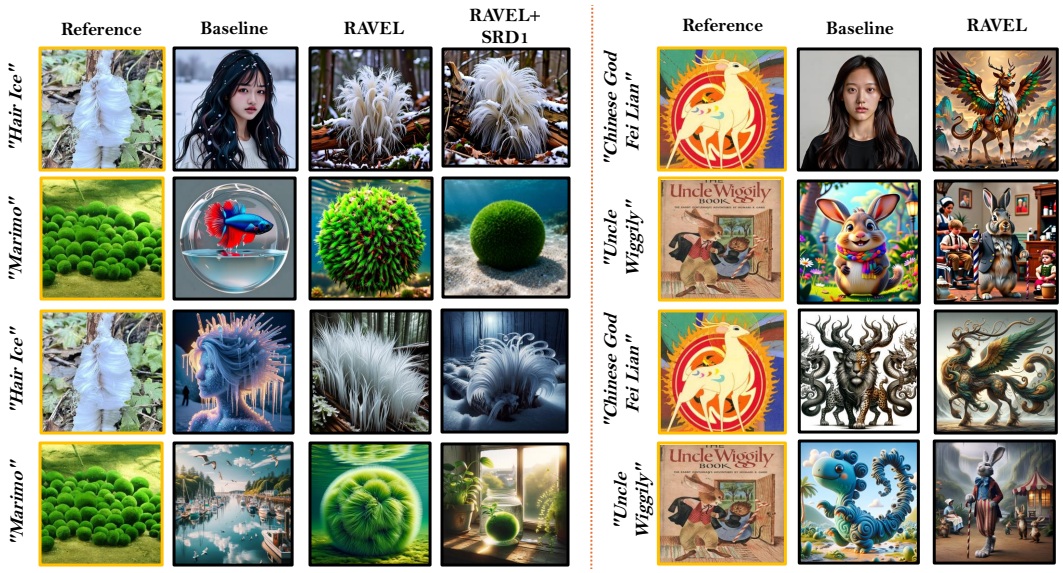

Figure 5: **Generation with GLM-Image and Janus-Pro.** GLM-Image (upper rows) and Janus-Pro (lower rows) on *Hair Ice*, *Marimo*, *Fei Lian*, and *Uncle Wiggily*. The GLM-Image examples also show SRD refinement. Reference images are for comparison only.

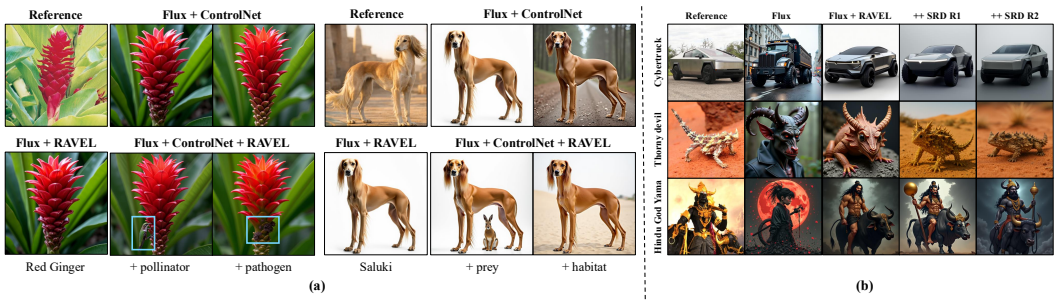

Figure 6: **Context-aware editing and self-correction.** (a) Knowledge-enriched ControlNet edits for *Red Ginger* and *Saluki*. (b) Successive SRD generations refine geometric, morphological, and symbolic attributes of *Cybertruck*, *Thorny Devil*, and *Yama*.

Image examples also show SRD correction; refinement trajectories for both generators appear in Appendix B.3.

4.3 Image Editing and Self-Correction

Context-aware image editing. In Fig. 6(a), knowledge-enriched ControlNet editing adds relevant elements to *Red Ginger* and *Saluki* scenes. The requested modifications concern relationships around the subject, such as introducing a pollinator or pathogen for the plant, or prey and habitat context for the dog. These examples show how the retrieved knowledge can guide the choice of added content alongside the editing pipeline's spatial conditioning. The editing task uses an input image, in contrast to the text-only input for rare-concept generation. Appendix D.1 provides further comparisons and quantitative results.

Iterative self-correction. Successive SRD generations in Fig. 6(b) improve the geometry of *Cybertruck*, morphological details of *Thorny Devil*, and symbolic features of *Yama*. The sequence illustrates how missing attributes can be addressed after the initial knowledge-conditioned generation. Each refinement uses feedback against the cached retrieved attributes to update the conditioning and generate a subsequent image.

Table 3: Human preference study (60 participants). Values indicate the percentage of forced-choice comparisons favoring each configuration for a given evaluation dimension.

Model Configuration	AA↑	VF↑	CR↑	IR↑
Flux (Baseline)	7.6%	9.7%	8.8%	10.3%
Flux + RAVEL	**92.4%**	**90.3%**	**91.2%**	**89.7%**
SDXL (Baseline)	9.9%	11.4%	9.6%	12.2%
SDXL + RAVEL	**90.1%**	**88.6%**	**90.4%**	**87.8%**
DALL-E 3 (Baseline)	8.4%	10.6%	9.2%	11.1%
DALL-E 3 + RAVEL	**91.6%**	**89.4%**	**90.8%**	**88.9%**

Self-correction performance. Table 4 compares RAVEL with two SRD refinement rounds against the base generators, SLD, and ReflectionFlow. Relative to the base generators, the combined pipeline improves text alignment, attribute evaluation, and reference-image similarity on every backbone, and it outperforms both self-correction baselines on all metrics while using fewer sequential refinement updates than ReflectionFlow. Relative to RAVEL without SRD (Table 1), refinement further increases CLIP-T, CLIP-I, DINO, BLIP-VQA, and SigLIP on every backbone. LPIPS improves for GLM-Image and Janus-Pro, whose initial RAVEL generations have the highest LPIPS, but rises slightly for FLUX, SDXL, and DALL-E 3, suggesting that regeneration can recover missing attributes at the cost of some perceptual drift when the initial image is already close to the reference.

Appendix E examines how these gains develop across refinement rounds. More images reach the ACS threshold as refinement continues, and most attributes verified in an earlier round remain present. Later rounds recover a smaller fraction of the missing attributes, suggesting diminishing returns from further refinement. Previously correct attributes can also be lost, and some images remain below the threshold. These results show why the evaluation tracks both the recovery of missing attributes and the retention of correct ones. ACS alone can hide changes within the attribute set: recovering several missing features can raise the score even when another feature is lost. Tracking both changes reveals whether refinement adds missing details while preserving earlier progress. These checks assess consistency with the retrieved attributes; they do not evaluate additional features outside that set.

4.4 User Study

We conduct a user study on the Prolific platform with 60 participants to compare RAVEL with SDXL, FLUX, and DALL-E 3 generators (see Table 3). Participants received textual concept descriptions and assessed Attribute Accuracy (AA), Visual Fidelity (VF), Context Relevance (CR) and Intent Representation (IR). Image order was randomized, and concepts were grouped by benchmark category for sampling. Participants strongly preferred RAVEL across all evaluated backbones and dimensions, with substantial agreement between participants. These judgments indicate that improved attribute depiction is accompanied by better contextual alignment and perceived visual quality. The textual descriptions give participants a common basis for assessing whether an image depicts the intended concept. Their preferences complement reference-image metrics, which measure resemblance to a selected depiction and may not capture every requested attribute. Participants sometimes preferred baseline images, particularly for concepts with limited attribute information or when visual appeal outweighed morphological accuracy. Please refer to the Appendix F.4 for more details about our user study.

4.5 Ablation Studies

We compare naive prompting, Vanilla RAG, Contrastive CoT, and Graph-Based RAG on SDXL, FLUX, and DALL-E 3 using the same textual sources and extracted concept information. Graph-based retrieval improves over flat Contrastive CoT across every evaluated backbone and metric, supporting the value of graph structure for conditioning (Appendix C.1). For example, linking a plant to its pollinator allows retrieval to collect the related entity's attributes and describe both entities and their relationship in the conditioning.

Table 4: **Self-Correction Performance Comparison.** Comparison of baseline generators, existing self-correction methods, and RAVEL with two SRD refinement rounds across several metrics and inference time.

Method	Rounds	CLIP-T↑	CLIP-I↑	DINO↑	VQA↑	LPIPS↓	SigLIP↑	Time (s)↓
Base Models								
Flux	0	0.187	0.418	0.287	0.228	0.580	0.063	62.5
SDXL	0	0.184	0.297	0.262	0.199	0.623	0.034	8.3
DALL-E 3	0	0.220	0.480	0.367	0.279	0.568	0.069	34.2
GLM-Image	0	0.232	0.533	0.391	0.297	0.612	0.052	71.6
Janus Pro	0	0.261	0.571	0.427	0.331	0.584	0.058	68.9
Self-Correction Methods								
SLD [35]	2–3	0.251	0.482	0.356	0.289	0.571	0.062	178.7
ReflectionFlow [41]	16	0.247	0.522	0.389	0.313	0.523	0.078	312.3
RAVEL with SRD (Ours)								
Flux + SRD R2	2	0.337	0.752	0.641	0.441	0.295	0.130	91.8
SDXL + SRD R2	2	0.321	0.733	0.598	0.428	0.330	0.109	27.3
DALL-E 3 + SRD R2	2	**0.368**	**0.780**	**0.693**	**0.482**	0.279	**0.139**	68.6
GLM-Image + SRD R2	2	0.334	0.716	0.626	0.439	0.352	0.093	92.9
Janus Pro + SRD R2	2	0.342	0.727	0.628	0.455	0.351	0.096	138.5

Adding RAVEL to ControlNet improves Attribute Accuracy and Context Relevance, supporting the use of retrieved knowledge to guide editing (Appendix D.1). We also compare GPT-4o, Llama-3.2, Qwen 2.5, and Mistral v0.3 for retrieval and conditioning construction. GPT-4o provides the strongest overall retrieval performance, while the open-source alternatives remain competitive (Appendix C.2).

5 Limitations

RAVEL depends on the quality and completeness of the attributes and relationships stored in its knowledge graph $\mathcal{G}$. Missing or inaccurate information can weaken the conditioning, although the graph can be inspected and updated to address such gaps. Even with accurate retrieval, textual descriptions may not fully convey subtle textures or material appearance, and complex or conflicting attributes can be difficult to depict together. Because RAVEL keeps pretrained model parameters fixed, its ability to resolve these errors remains limited by how well the generator follows the text conditioning.

6 Conclusion

We present **RAVEL**, a training-free framework that retrieves concept attributes and relationships from a knowledge graph to improve semantic grounding in text-to-image generation. This information guides the generator through text conditioning. Across the evaluated cultural artifacts, biological entities, and other rare concepts, RAVEL improves the depiction of defining visual attributes and their relationships. The same retrieved information supports context-aware editing, while SRD uses VLM feedback to address missing attributes in subsequent generations. Together, these capabilities show how external concept information can guide diffusion and autoregressive image generators without model fine-tuning.

7 Acknowledgment

This research is supported by the National Science Foundation under Grant No. 2543524.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 20621–20637. Curran Associates, Inc., 2022.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[4] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022.

[5] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[6] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[7] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023.

[8] Yutong Feng, Biao Gong, Di Chen, Yujun Shen, Yu Liu, and Jingren Zhou. Ranni: Taming text-to-image diffusion for accurate instruction following. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4744–4753, June 2024.

[9] 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. *arXiv preprint arXiv:2208.01618*, 2022.

[10] Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. Llm blueprint: Enabling text-to-image generation with complex and detailed prompts. *arXiv preprint arXiv:2310.10640*, 2023.

[11] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control.(2022). *URL https://arxiv.org/abs/2208.01626*, 3, 2022.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] 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 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 78723–78747. Curran Associates, Inc., 2023.

[14] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[15] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[18] 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. *arXiv preprint arXiv:2305.13655*, 2023.

[19] Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–23, 2022.

[20] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[21] Chen-Yi Lu, Shubham Agarwal, Md Mehrab Tanjim, Kanak Mahadik, Anup Rao, Subrata Mitra, Shiv Kumar Saini, Saurabh Bagchi, and Somali Chaterji. Recon: Training-free acceleration for text-to-image synthesis with retrieval of concept prompt trajectories. In *European Conference on Computer Vision*, pages 288–306. Springer, 2024.

[22] Yuanhuiyi Lyu, Xu Zheng, Lutao Jiang, Yibo Yan, Xin Zou, Huiyu Zhou, Linfeng Zhang, and Xuming Hu. Realrag: Retrieval-augmented realistic image generation via self-reflective contrastive learning. *arXiv preprint arXiv:2502.00848*, 2025.

[23] Neo4j Inc. Neo4j: The graph database platform, 2023. Accessed: November 14, 2024.

[24] OpenAI. DALL-E 3. `https://openai.com/index/dall-e-3/`, 2023.

[25] OpenAI. Introducing GPT-5.2. `https://openai.com/index/introducing-gpt-5-2/`, December 11 2025. Release announcement.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[27] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[29] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4695–4703, 2024.

[30] Priyanka Sen, Sandeep Mavadia, and Amir Saffari. Knowledge graph-augmented language models for complex question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8, 2023.

[31] Rotem Shalev-Arkushin, Rinon Gal, Amit H Bermano, and Ohad Fried. Imagerag: Dynamic image retrieval for reference-guided image generation. *arXiv preprint arXiv:2502.09411*, 2025.

[32] Yichen Sun, Zhixuan Chu, Zhan Qin, and Kui Ren. Prompt-consistency image generation (pcig): A unified framework integrating llms, knowledge graphs, and controllable diffusion models. *arXiv preprint arXiv:2406.16333*, 2024.

[33] Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395, 2024.

[34] Sam Witteveen and Martin Andrews. Investigating prompt engineering in diffusion models. *arXiv preprint arXiv:2211.15462*, 2022.

[35] Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6327–6336, 2024.

[36] Z.AI Team. GLM-Image: Autoregressive image generation with hybrid autoregressive and diffusion decoder architecture. `https://huggingface.co/zai-org/GLM-Image`, 2026. Accessed: 2026.

[37] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[38] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[40] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, Caiming Xiong, and Ran Xu. Hive: Harnessing human feedback for instructional visual editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9026–9036, June 2024.

[41] Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15329–15339, 2025.

Table of Contents

A Computational Cost and Deployment **1**

B Additional Qualitative Results **2**

B.1 Rare-Concept Generation Across Domains 2

B.2 Mythological and Literary Concepts 2

B.3 SRD Refinement with Autoregressive Models 3

C Ablation Studies and Prompt Enhancement Baselines **3**

C.1 Retrieval and Text Conditioning Ablations 3

C.2 Choice of Language Model 4

C.3 Comparison with LLM/VLM Prompt Enhancement 4

D External Benchmarks and Image Editing **5**

D.1 Context-Aware Image Editing 5

E RAVEL Inference and SRD Evaluation **6**

E.1 Unified Inference Procedure 6

E.2 Reaching the ACS Threshold 6

E.3 Attribute Retention and Recovery 7

F Benchmark Construction and Evaluation Protocols **8**

F.1 Benchmark Composition 8

F.2 Benchmark Annotations and Evaluation Independence 8

F.3 Reference Image Construction 9

F.4 Human Preference Study 9

G Discussion and Broader Impact **10**

H Knowledge Graph Construction and Validation **10**

H.1 Graph Schema and Construction 10

H.2 Domain-Specific Instantiations 11

H.3 Quality Assurance and Knowledge Validation 11

H.4 Retrieval and Contrastive Prompt Synthesis 12

H.5 Knowledge Graph Extraction System Prompt 12

A Computational Cost and Deployment

Computational cost. RAVEL retrieves attributes and relationships from the knowledge graph once per request. A large language model (LLM) combines them with the request to construct text conditioning, which the generator uses to produce the initial image. Without SRD, this requires a single call to the image generator.

With SRD, a vision-language model checks each image against the same visual attributes retrieved for the target entities. The Attribute Coverage Score (ACS) is the fraction of those attributes it judges

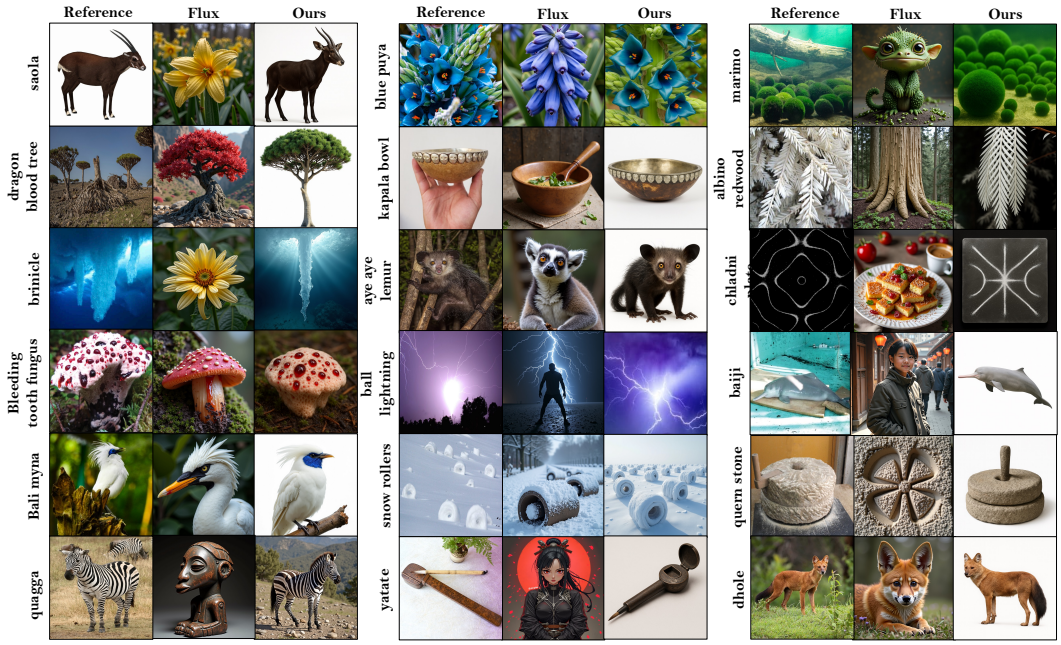

Figure 7: **Generation of rare species, physical phenomena, and historical artifacts.** Within each group, FLUX generates from the original request (center) or from text conditioning that RAVEL constructs using the request and retrieved attributes and relationships (right). The reference image (left) is shown only for comparison.

present in the current image. If ACS is below $\tau = 0.85$ and another update is available, SRD uses feedback about the missing attributes to revise the conditioning. Each update reuses the retrieved information and adds one generation and one verification step. The limit of $K_{\max} = 3$ updates allows up to four image-generation calls, including the initial call. Table 4 reports runtime for two SRD updates alongside results for the base generators, SLD, and ReflectionFlow. Appendix E examines how often images reach the threshold within the full budget.

Deployment. We store the knowledge graph in Neo4j [23] and use GPT-4o [1] to identify target entities and construct text conditioning from the retrieved attributes and relationships. Appendix C.2 compares alternative LLMs for these tasks. We pass the conditioning through each generator's existing text interface, keeping all pretrained parameters fixed. We run local generators on a single NVIDIA A100 and access hosted generators through their provider APIs.

B Additional Qualitative Results

B.1 Rare-Concept Generation Across Domains

Figure 7 compares FLUX with and without RAVEL across species such as the *saola*, physical phenomena such as *ball lightning*, and historical artifacts such as the *yatate*. For a *brinicle*, FLUX depicts a flower, while FLUX with RAVEL shows ice extending downward through the water. For a *Chladni plate*, RAVEL depicts a geometric line pattern where FLUX depicts a plate of food. The *kapala bowl* illustrates a subtler difference. Both outputs depict a bowl, but only the RAVEL output includes the decorated rim visible in the reference.

B.2 Mythological and Literary Concepts

Figure 8 compares FLUX with and without RAVEL on figures from Indian, Chinese, and Greek mythology and on literary characters. For *Gandabherunda*, FLUX depicts a single-headed dragon, while FLUX with RAVEL shows the two-headed bird in flight. In the Greek mythology examples, FLUX renders *Iris* as a flower, while the RAVEL output depicts the winged goddess holding a staff.

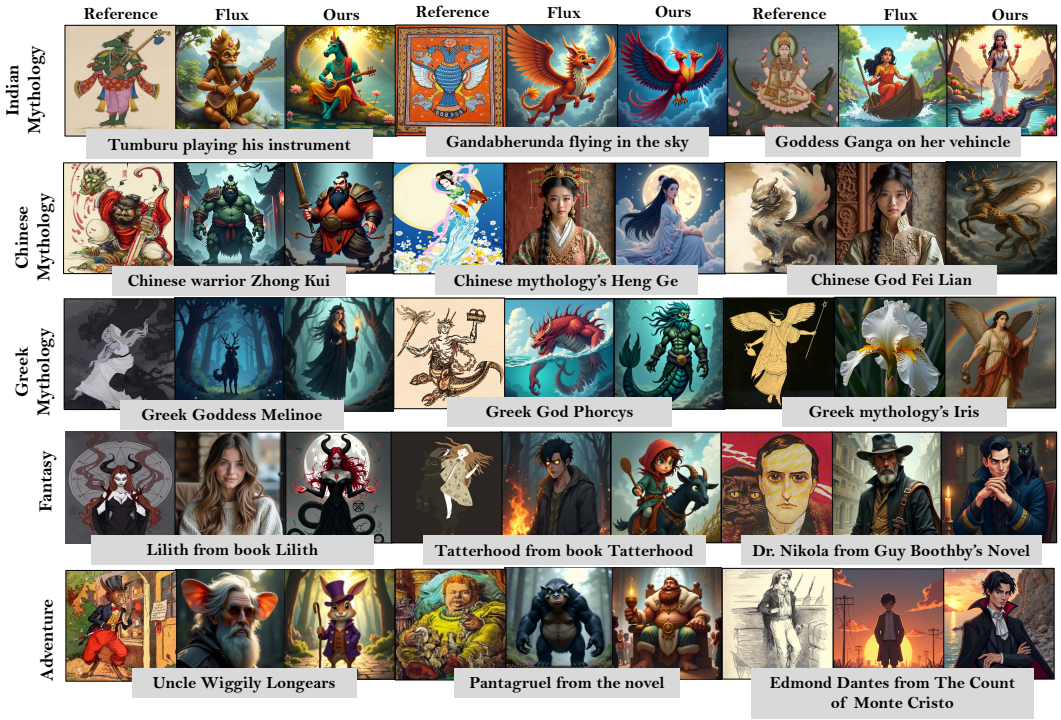

Figure 8: **Mythological and literary characters.** Rows 1–3 show Indian, Chinese, and Greek mythology, respectively. The literary examples from Project Gutenberg are grouped by genre, with fantasy in row 4 and adventure in row 5. Each group shows, from left to right, an illustrative reference, FLUX, and FLUX with RAVEL.

The literary examples show *Tatterhood* riding a goat and *Uncle Wiggily* as a rabbit in a top hat, details visible in both the RAVEL outputs and the reference illustrations.

B.3 SRD Refinement with Autoregressive Models

Figure 9 shows successive SRD updates for the two generators compared in Fig. 5. GLM-Image combines autoregressive modeling with diffusion, while Janus-Pro generates images autoregressively. For *Bleeding Tooth Fungus*, red droplets appear before either generator captures the pale, irregular surface shown in the reference. Refinement can also introduce errors. In the Janus-Pro sequence for *Yama*, the first update corrects the skin color but adds extra arms and heads. The final round corrects these errors and reaches the ACS threshold. Appendix E examines how often SRD recovers missing attributes and retains those already present.

C Ablation Studies and Prompt Enhancement Baselines

C.1 Retrieval and Text Conditioning Ablations

Figure 10 compares four ways to construct conditioning from the same textual sources and extracted concept information. Naive prompting supplies the full text available for each request. Vanilla RAG retrieves relevant passages, while Contrastive CoT uses a flat list of attributes to distinguish the target from similar concepts. Graph-Based RAG also uses typed relationships to select the entities and attributes included in the conditioning.

On FLUX, SDXL, and DALL-E 3, Graph-Based RAG consistently outperforms flat Contrastive CoT across all three evaluated dimensions: Attribute Accuracy, Context Relevance, and Intent Representation. The improvements are particularly pronounced in Context Relevance, supporting the role of structured relationships in grounding entities within their intended semantic and contextual settings.



Figure 9: **SRD refinement with GLM-Image (left) and Janus-Pro (right).** For each model, the initial RAVEL output is followed by successive SRD rounds. Annotations identify missing or recovered attributes and mark the round that meets the ACS threshold. Reference images are shown only for comparison.

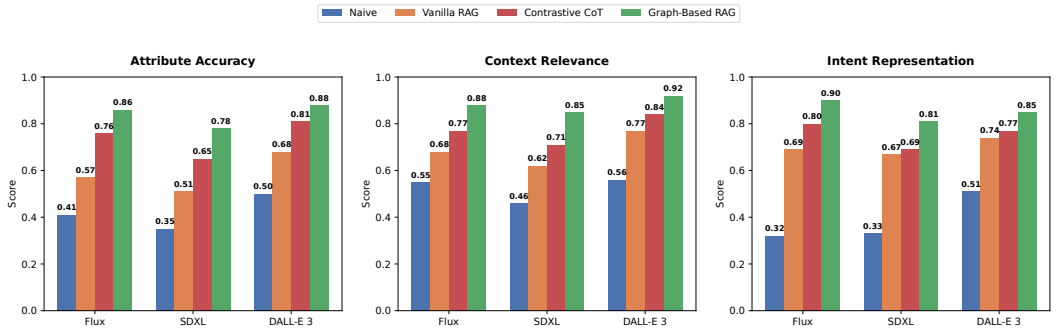

Figure 10: **Ablation of knowledge-grounded generation.** Comparison of Naive prompting, Vanilla RAG, Contrastive CoT, and Graph-Based RAG across FLUX, SDXL, and DALL-E 3. All configurations use the same source corpus and extracted concept information. Graph-based retrieval outperforms flat Contrastive CoT across all nine backbone–metric comparisons, with absolute improvements of 0.070–0.140 (mean +0.103).

C.2 Choice of Language Model

Figure 11 compares GPT-4o, Llama-3.2, Qwen 2.5, and Mistral v0.3 for graph retrieval and conditioning construction. GPT-4o achieves the highest retrieval accuracy, context precision, and context recall, followed by Llama-3.2 on all three measures. These two models also score highest on faithfulness and lowest on hallucination. The models are closer on contrastive effectiveness, where Llama-3.2 and Mistral v0.3 both approach GPT-4o.

C.3 Comparison with LLM/VLM Prompt Enhancement

We compare RAVEL with prompt enhancement techniques using GPT-5.2 and Claude Haiku 4.5. Figure 4 in the main paper illustrates concept completion, CoT expansion, search augmentation, and

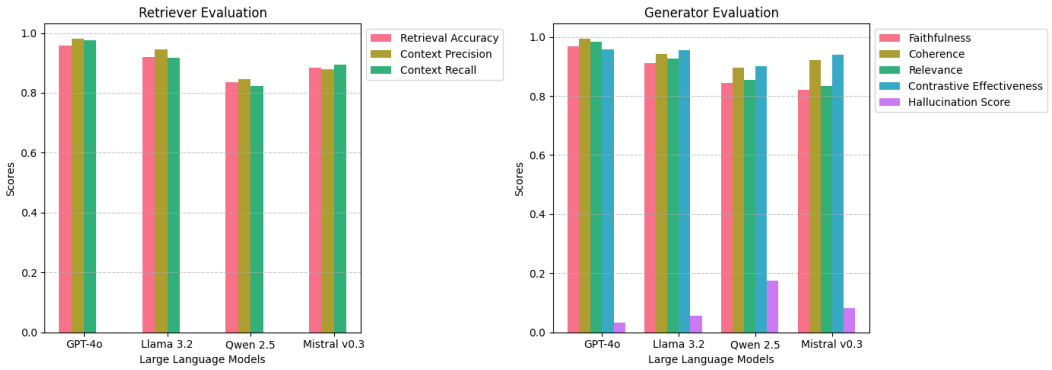

Figure 11: **Effect of language model choice on retrieval and generation.** The left panel reports retrieval accuracy, context precision, and context recall. The right panel reports faithfulness, coherence, relevance, contrastive effectiveness, and hallucination score. Lower hallucination scores are better.

Table 5: **Prompt enhancement on FLUX and GLM-Image.** DINO and VQA denote DINOv2 and BLIP-VQA. Coverage and Hallucination report attribute coverage and hallucination rate. These comparisons were run separately from the main cross-backbone evaluation, so aggregate baseline scores can differ slightly between the two settings.

Method	**CLIP-T↑**	**CLIP-I↑**	**DINO↑**	**VQA↑**	**Coverage↑**	**Hallucination↓**
FLUX Backbone						
FLUX (Base)	0.187	0.418	0.287	0.228	27.4%	62%
GPT CoT	0.268	0.531	0.401	0.356	61%	29%
GPT Iterative	0.289	0.558	0.428	0.378	68%	22%
GPT + Search	0.301	0.589	0.451	0.391	74%	18%
Claude CoT	0.259	0.519	0.389	0.344	58%	31%
Claude Iterative	0.278	0.544	0.413	0.364	61%	28%
Claude + Search	0.291	0.571	0.438	0.382	68%	23%
RAVEL (Ours)	**0.331**	**0.681**	**0.582**	**0.422**	**87.5%**	**11%**
GLM-Image Backbone						
GLM (Base)	0.229	0.528	0.383	0.290	41.4%	33.2%
CoT	0.237	0.574	0.436	0.364	57.8%	26.9%
Iterative ($K = 3$)	0.298	0.581	0.447	0.374	64.0%	21.0%
Search-Enabled	0.311	0.607	0.471	0.399	65.8%	27.6%
RAVEL + SRD	**0.331**	**0.712**	**0.622**	**0.441**	**83.8%**	**11.3%**

iterative VQA refinement with up to five updates. Within each comparison, the methods use the same image generator and consistent prompt formatting.

Table 5 compares CoT, search, and iterative refinement on FLUX and GLM-Image. Among the FLUX prompt enhancement baselines, search-enabled enhancement gives both GPT-5.2 and Claude Haiku 4.5 their highest attribute coverage and lowest hallucination rates. While search also gives the highest baseline coverage on GLM-Image, iterative refinement has the lowest baseline hallucination rate. RAVEL on FLUX and RAVEL with SRD on GLM-Image each improve coverage and reduce hallucinations relative to every baseline on the same generator.

D External Benchmarks and Image Editing

D.1 Context-Aware Image Editing

Table 6 compares the ControlNet editing pipeline with and without RAVEL. RAVEL incorporates retrieved attributes and relationships into the editing instruction, while ControlNet provides spatial control. The largest gains are in Attribute Accuracy and Context Relevance, indicating that the edits

Table 6: **Effect of RAVEL on ControlNet editing.** The columns report Attribute Accuracy (AAI), Context Relevance (CRS), Visual Fidelity (VFS), and Intent Representation (IRM).

Method	AAI↑	CRS↑	VFS↑	IRM↑
ControlNet	0.317	0.284	0.334	0.290
ControlNet + RAVEL	**0.897**	**0.859**	**0.867**	**0.855**

Table 7: **Instruction following and input preservation in image editing.** DINO and VQA denote DINOv2 and BLIP-VQA. CLIP-I and DINOv2 measure similarity to the original input. LPIPS measures perceptual distance from that input, with smaller values indicating less change.

Method	CLIP-T↑	CLIP-I↑	DINO↑	VQA↑	LPIPS	SigLIP↑
Instruction-Based Editing						
InstructPix2Pix [3]	0.291	0.901	0.881	0.369	0.089	0.101
Prompt2Prompt [11]	0.276	0.912	0.890	0.372	0.083	0.096
DiffEdit [6]	0.303	0.919	0.902	0.379	0.076	0.107
Model-Based Editing						
ControlNet	0.318	**0.981**	**0.939**	0.368	0.064	0.110
HIVE [40]	0.302	0.927	0.911	0.376	0.231	0.116
GenArtist [33]	0.308	0.934	0.916	0.377	0.227	0.118
ControlNet + RAVEL (Ours)	**0.358**	0.940	0.922	**0.429**	0.284	**0.128**

better capture the requested features and their setting. We additionally report improvements in Visual Fidelity and Intent Representation.

Table 7 compares how well editing methods follow the instruction and how closely their outputs resemble the input image. ControlNet with RAVEL scores highest on instruction alignment (CLIP-T and SigLIP) and attribute verification (BLIP-VQA). Standard ControlNet produces the edits most similar to the input according to CLIP-I, DINOv2, and LPIPS. Read together, these measures show that RAVEL makes larger visual changes while following the requested edit more closely.

Figure 12 shows how the same editing requests affect the subject and the surrounding scene. For *Shiva* (row 1), the request to add *Vasuki* produces a coiled snake at Shiva's feet with RAVEL. For *Ganesha* (row 2), the request to show his abode changes the garden to a mountain setting. For *Hanuman* (row 3), ControlNet with RAVEL depicts his mace in the weapon edit and a green plant in the *Sanjeevini* edit.

E RAVEL Inference and SRD Evaluation

E.1 Unified Inference Procedure

RAVEL first combines the request with attributes and relationships retrieved from the graph to construct the text conditioning for the initial image $\mathcal{I}_0$ (Algorithm 1). When SRD is disabled, the procedure returns $\mathcal{I}_0$ at this point.

For SRD, let $\mathcal{A}_{\text{ret}} = \{a_1, \ldots, a_N\}$ be the visual attributes retrieved for the target entities, with $N > 0$. A VLM checks every image against this fixed set, assigning $b_{t,i} = 1$ when it judges a_i present in $\mathcal{I}_t$ and 0 otherwise. The Attribute Coverage Score (ACS_t) is the fraction judged present. If ACS is below $\tau = 0.85$ and another update is available, SRD uses the missing attributes and the cached retrieval to revise the conditioning and generate the next image. It returns the current image as soon as ACS reaches the threshold or after $K_{\max} = 3$ updates following the initial generation.

E.2 Reaching the ACS Threshold

Table 8 shows when examples in the 500-image evaluation reach the ACS threshold. About half meet it at initial generation. The first two SRD updates account for most of the subsequent gain, while a smaller share reaches the threshold on the final update.



Figure 12: **Context-aware editing with ControlNet.** Each block begins with the same input image generated by FLUX with RAVEL. The remaining columns show edits by standard ControlNet (left block) or ControlNet with RAVEL (right block). Labels beneath the images give the requested changes.

Algorithm 1 RAVEL generation with optional SRD refinement

Require: User request $\mathcal{P}$, knowledge graph $\mathcal{G}$, SRD setting (on/off), ACS threshold τ, maximum updates $K_{\max}$
Ensure: Image returned for the request

$(\mathcal{C}_{\mathcal{P}}, \mathcal{A}_{\text{ret}}) \leftarrow \textsc{RetrieveKnowledge}(\mathcal{P}, \mathcal{G})$
$\mathcal{P}_0 \leftarrow f_{\text{syn}}(\mathcal{P}, \mathcal{C}_{\mathcal{P}})$
$\mathcal{I}_0 \leftarrow \textsc{Generate}(\mathcal{P}_0)$
if SRD is off **then**
  return $\mathcal{I}_0$
end if
$N \leftarrow |\mathcal{A}_{\text{ret}}|$ (SRD assumes $N > 0$)
for $t = 0, \ldots, K_{\max}$ **do**
  $\mathbf{b}_t \leftarrow \textsc{VerifyAttributes}(\mathcal{I}_t, \mathcal{A}_{\text{ret}})$
  $\text{ACS}_t \leftarrow \frac{1}{N} \sum_{i=1}^{N} b_{t,i}$
  if $\text{ACS}_t \geq \tau$ or $t = K_{\max}$ **then**
    return $\mathcal{I}_t$
  end if
  $\mathcal{A}_{\text{miss},t} \leftarrow \{a_i \in \mathcal{A}_{\text{ret}} : b_{t,i} = 0\}$
  $\mathcal{P}_{t+1} \leftarrow f_{\text{refine}}(\mathcal{P}_t, \mathcal{A}_{\text{miss},t}, \mathcal{C}_{\mathcal{P}})$
  $\mathcal{I}_{t+1} \leftarrow \textsc{Generate}(\mathcal{P}_{t+1})$
end for

E.3 Attribute Retention and Recovery

We compare the VLM's judgments for each attribute before and after an SRD update. Among attributes judged present before the update, *retention* is the fraction still judged present afterward and *regression* is the fraction judged missing. Among attributes previously judged missing, *recovery* is the fraction judged present after the update. Each transition includes only requests that proceed to that round.

Table 9 shows that most attributes judged present remain present across an update. Among the requests that still need refinement, later rounds lose a smaller share of previously verified attributes but also recover a smaller share of those still missing.

Table 8: **Reaching the ACS threshold during SRD.** Each row gives the percentage of all evaluation examples that have reached $\mathrm{ACS} \geq 0.85$ by the end of that stage.

Refinement stage	Reached threshold by this stage
Initial RAVEL generation	51.1%
After R1	64.5%
After R2	77.6%
After R3	84.3%

Table 9: **Attribute changes between successive SRD rounds.** Entries give the mean and sample standard deviation of the rates computed separately for the five image generators, in percent.

Transition	Retention ↑	Regression ↓	Recovery ↑
Initial → R1	$(96.4 \pm 1.5)\%$	$(3.6 \pm 1.5)\%$	$(65.8 \pm 6.9)\%$
R1 → R2	$(97.5 \pm 1.0)\%$	$(2.5 \pm 1.0)\%$	$(57.8 \pm 7.4)\%$
R2 → R3	$(98.1 \pm 0.8)\%$	$(1.9 \pm 0.8)\%$	$(43.6 \pm 8.6)\%$

F Benchmark Construction and Evaluation Protocols

This section details the construction of our rare-concept benchmarks, evaluation annotations, reference-image curation, and automated and human evaluation protocols.

F.1 Benchmark Composition

Our benchmark suite comprises 2,000 prompts spanning 600 distinct rare concepts across more than 15 categories. The three benchmarks target complementary challenges: cultural and symbolic composition, fine-grained rare-concept grounding, and narrative generalization.

MythoBench. MythoBench contains 500 character prompts derived from Wikipedia and cross-referenced with encyclopedic and public-domain sources. Prompts are stratified into three difficulty tiers: *simple* (≤ 3 attributes), *composite* (4–6 attributes), and *relational* (≥ 7 attributes with explicit relational dependencies). The benchmark evaluates the composition of morphological, cultural, and symbolic attributes in visually coherent representations.

Rare-Concept-1K. Rare-Concept-1K contains 1,000 prompts distributed equally across five under-represented categories: obsolete technologies, rare creatures, cultural artifacts, architectural objects, and uncommon tools. Each category comprises 200 prompts constructed from public sources, including Wikipedia, taxonomic databases, and museum archives. Prompts emphasize distinctive attributes that cannot be reliably inferred from entity names alone, requiring explicit contextual grounding.

NovelBench. NovelBench contains 500 prompts derived from public-domain literary sources, including Project Gutenberg. Prompts are paraphrased from textual descriptions rather than copied verbatim. The benchmark assesses zero-shot narrative generalization to literary characters and entities lacking widely established visual representations.

F.2 Benchmark Annotations and Evaluation Independence

Each benchmark entry contains a canonical concept name, domain category, structured attribute set (visual, functional, relational, and contextual), contrastive and composite prompts, a binary ground-truth attribute vector, and an SRD target set $\mathcal{T}$ for iterative verification.

To maintain independence between generation and evaluation, ground-truth attribute vectors were prepared through a separate annotation pass from runtime knowledge-graph construction. Annotations were manually checked against source documentation and held fixed across all evaluated methods. RAVEL retrieves prompt-conditioned attributes from the knowledge graph during inference, whereas evaluation uses the independently established benchmark targets.

This separation prevents direct reuse of runtime-retrieved attributes as evaluation labels. However, because benchmark annotations and graph attributes may originate from overlapping source corpora and similar attribute categories, some schema-level alignment remains possible. We therefore complement attribute-based evaluation with perceptual metrics, external benchmarks, and human judgments.

The benchmark suite, annotations, prompt templates, ground-truth vectors, and evaluation scripts are prepared for public release under CC BY 4.0, together with knowledge-graph snapshots and the construction prompt provided in Appendix H.5.

F.3 Reference Image Construction

For image-based metrics (CLIP-I, DINOv2, and LPIPS), we construct surrogate references representing canonical depictions of each concept. We collect 3-10 candidates, retain those satisfying at least 80% of documented defining attributes, and select the highest-scoring candidate. The same reference image is used for all methods evaluated on a concept and is never provided during generation. For editing, image-similarity metrics are computed relative to the original input image.

Reference Sources. Images are collected from domain-relevant repositories:

- **Mythology and Literature:** Wikimedia Commons (`https://commons.wikimedia.org`), British Museum (`https://www.britishmuseum.org/collection`), Sanskrit Documents (`https://sanskritdocuments.org`), Theoi (`https://www.theoi.com`), Exotic India Art (`https://www.exoticindiaart.com`), Project Gutenberg (`https://www.gutenberg.org`), and Internet Archive (`https://archive.org`).
- **Cultural Artifacts, Architecture, and Fashion:** Metropolitan Museum of Art (`https://www.metmuseum.org/art/collection`), Smithsonian Open Access (`https://www.si.edu/openaccess`), Europeana (`https://www.europeana.eu`), and Victoria and Albert Museum (`https://collections.vam.ac.uk`).

These references provide common comparison targets across baselines rather than generation-time guidance.

F.4 Human Preference Study

We conducted a controlled study on Prolific with 60 participants (platform approval rate $> 95\%$). All participants evaluated the same set of 25 concepts, while baseline-RAVEL image-pair assignments from three generative backbones (FLUX, SDXL, and DALL-E 3) and left/right presentation were randomized across participants. Each participant completed 25 paired comparisons, assessing Attribute Accuracy, Visual Fidelity, Context Relevance, and Intent Representation. Preference percentages are computed separately for each backbone and evaluation dimension from the corresponding valid judgments. Each concept was evaluated by at least three independent participants. Three attention-check trials were embedded in the study; incomplete or low-effort responses were filtered and replaced, while participants were not excluded solely on the basis of attention-check outcomes.

Results. Across all three backbones and four evaluation dimensions, participants preferred RAVEL-enhanced outputs in an average of **90.10%** of comparisons. Preferences remained consistently high across biology (90.2%), mythology (93.1%), cultural artifacts (91.8%), architectural objects (90.7%), and uncommon tools (89.4%). Inter-rater agreement was substantial, with Fleiss' κ values of 0.71 (AAI), 0.68 (VFS), 0.74 (CRS), and 0.66 (IRM).

Failure Cases. Baseline outputs were occasionally preferred, particularly for concepts with sparse attribute coverage or when visual aesthetics outweighed strict morphological accuracy. These cases highlight the distinction between perceived visual quality and attribute-grounded fidelity, motivating the use of complementary evaluation criteria.

G Discussion and Broader Impact

RAVEL addresses a fundamental challenge in rare-concept generation: even strong T2I and VLM systems struggle with concepts whose visual priors are weak, absent, or confounded. Mythological and literary entities often have diverse interpretations across cultures, regions, and artistic traditions, while rare biological species and cultural artifacts require precise binding of fine-grained structural, symbolic, and relational attributes that are sparsely represented in training data. Consequently, models may collapse toward familiar high-frequency visual priors, overemphasize salient features, or generate incoherent composites from conflicting contexts. These failures are not reliably resolved by stronger models, longer prompts, or unstructured retrieval alone. By incorporating structured, user-validated relational knowledge at inference time, RAVEL makes defining attributes and contextual relationships explicit, enabling more faithful and controllable visual grounding without modifying model parameters. Importantly, the framework does not resolve ambiguities in source knowledge; generated outputs remain dependent on the selected representations and should not be considered universally authoritative depictions.

Scientific and Educational Utility. Attribute-grounded generation may support education, scientific communication, and museum applications involving rare species, historical artifacts, or sparsely documented phenomena. Such visualizations can supplement limited reference material but should be interpreted as informed approximations rather than definitive reconstructions.

Transparency and Control. RAVEL's explicitly represented attributes and relationships allow practitioners to inspect, verify, and modify the knowledge used for generation. This provides greater control over conditioning, particularly in culturally sensitive or domain-specific settings where the provenance and interpretation of visual attributes matter.

Risks and Responsible Use. Knowledge sources may contain incomplete, biased, or contested cultural and historical interpretations that generated outputs could perpetuate. Plausible depictions of culturally significant entities may also be misused to create misleading representations. Since RAVEL does not independently establish source correctness, responsible applications involving living traditions or sacred iconography should account for contextual differences and involve relevant community perspectives.

Computational Considerations. As a training-free framework, RAVEL avoids additional model training, although retrieval, prompt construction, and optional SRD refinement introduce inference overhead. Its modular design allows practitioners to balance refinement depth, computational cost, and latency.

Benchmarks and Evaluation. MythoBench, Rare-Concept-1K, and NovelBench complement existing benchmarks by evaluating attribute grounding and compositional fidelity across rare cultural, biological, and literary concepts. They provide structured testbeds for studying long-tail generation failures without claiming exhaustive coverage of these domains.

H Knowledge Graph Construction and Validation

RAVEL constructs a directed, multi-relational knowledge graph (KG) from textual sources to represent rare concepts through explicit attributes and semantic relationships. We describe the graph schema, construction and validation procedures, retrieval strategy, and knowledge-extraction prompt.

H.1 Graph Schema and Construction

Given a source corpus $\mathcal{D}$, RAVEL constructs a knowledge graph $G = (V, R, E)$, where V denotes entities, R typed relationships, and E entity-associated attributes. Each entity is characterized by four attribute groups: visual (appearance and morphology), functional (roles and behavior), relational (associations with other entities), and contextual (historical, geographic, or narrative information).

Construction proceeds in two stages. First, an LLM extracts structured attributes from source documents and instantiates entity-centric subgraphs. Entity and attribute nodes are connected

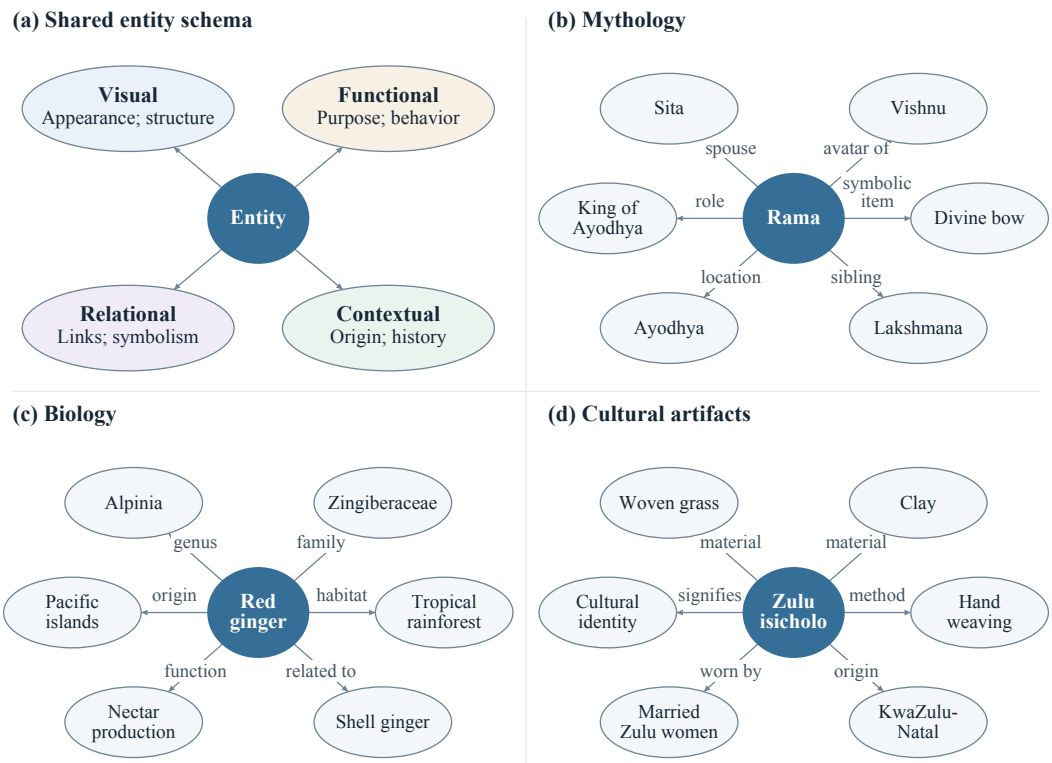

Figure 13: **RAVEL knowledge-graph schema.** The general schema organizes entities and typed relationships into a domain-adaptable representation. The examples illustrate instantiations across mythology, biology, and cultural artifacts.

through typed relationships such as `HAS_APPEARANCE` and `SYMBOLIC_ITEM`, and stored in Neo4j using generated Cypher queries. Second, independently constructed subgraphs are linked through inter-entity relationships spanning ecological, familial, functional, and cultural associations.

Relationships are represented as directed triples (v_i, r, v_j), with reciprocal edges instantiated using the same relation for symmetric associations or an inverse relation for asymmetric associations. For example, `POLLINATED_BY` and `POLLINATES` form an inverse pair. This representation supports traversal across related entities while preserving their distinct attributes and contextual relationships.

H.2 Domain-Specific Instantiations

The graph schema supports different knowledge domains without requiring domain-specific architectural changes. In biology, *Red Ginger* is represented through morphological, taxonomic, and ecological relationships, including `POLLINATED_BY` and `FOUND_IN_HABITAT`. Cultural artifacts such as *Zulu Isicholo* are characterized by materials, construction methods, and relationships such as `WORN_WITH` and `SIGNIFIES`. For mythology, entities such as *Rama* are linked through symbolic and narrative relations, including `HAS_SPOUSE` and `AVATAR_OF`. These examples illustrate how relational grounding accommodates domain-specific knowledge within a unified graph representation.

H.3 Quality Assurance and Knowledge Validation

The KG is constructed from publicly documented sources, including encyclopedic records, biological databases, museum collections, and historical or literary texts. Extracted attributes and relationships undergo source cross-referencing, automated consistency checks, and manual review. Automated checks identify duplicate entities, invalid relationship types, and missing reciprocal edges.

Manual Knowledge-Graph Audit. We conducted a complete manual audit covering all 600 core entities and 2,133 directed relationships. Initial extraction accuracy was 95.83% for entities and 92.78% for relationships. Identified errors were corrected before finalizing the evaluated graph. This audit measures knowledge-extraction quality independently of downstream image-generation performance.

Conflicting Sources and Contextual Variants. Certain concepts, particularly cultural and mythological entities, admit multiple documented interpretations. When sources describe different regional or narrative manifestations, their attributes are retained with the corresponding contextual information rather than combined into a single representation. Discrepancies are flagged for source cross-referencing and manual resolution where necessary.

During inference, RAVEL can condition retrieval on the narrative or cultural context specified in the prompt. Alternative documented manifestations remain distinguishable within the graph. The resulting depiction nevertheless depends on the selected sources and should not be treated as a uniquely authoritative representation.

H.4 Retrieval and Contrastive Prompt Synthesis

At inference time, RAVEL identifies the entities and relational cues specified in the user prompt and retrieves their corresponding subgraphs through Cypher queries. The default retrieval uses a one-hop neighborhood, including immediate attributes and typed relationships. Multi-hop traversal is used when prompts require information about linked entities, such as familial or ecological associations.

The retrieved context is cached and organized into visual, functional, relational, and contextual attributes. Contrastive Chain-of-Thought (CoT) prompt synthesis emphasizes features distinguishing the target concept from common semantic neighbors. For example, an "upright torch-like, not hanging" descriptor can disambiguate a rare floral structure from generic category-level representations. The enriched prompt conditions the selected T2I backbone without modifying its parameters.



Figure 14: **Attribute coverage across semantic categories.** Comparison of graph-based retrieval and vector RAG over eight attribute groups.

Attribute Coverage. Figure 14 compares the attribute coverage achieved by graph-based retrieval with conventional vector RAG across eight semantic categories. RAVEL achieves **87.5%** average attribute coverage, compared with **30.4%** for vector RAG, indicating that structured retrieval recovers a broader range of concept-specific information.

H.5 Knowledge Graph Extraction System Prompt

The following system prompt specifies the extraction schema used to transform source text into structured entity attributes. It provides domain-specific extraction guidelines while retaining a common representation across concept categories.

System Prompt: Knowledge Graph Extraction

Instruction
You are an assistant tasked with building a knowledge graph for rare and underrepresented concepts across diverse domains including mythology, biology, cultural artifacts, historical objects, and fiction. Given information from authoritative sources (Wikipedia, taxonomic databases, museum catalogs, verified texts) about the entity `{entity_name}` from the `{domain}`, analyze the source text to extract structured attributes according to the domain-specific guidelines and return the information in JSON format.

Guidelines for extracting information:

1. **name**: Commonly known entity name or designation.
2. **domain**: Domain category (mythology, biology, cultural_artifact, architecture, technology, fashion, etc.).
3. **type**: Specific type within the domain (e.g., deity, plant, animal, clothing, tool).
4. **origin**: Geographic, cultural, or historical origin.
5. **primary_sources**: Authoritative references documenting the entity.
6. **alternative_names**: Other names, titles, scientific nomenclature, or regional variations.

Domain-Specific Attribute Guidelines:

- **Mythology/Historical Figures:** appearance_traits, personality_traits, symbolic_items, dwelling, powers_abilities, weaknesses, relationships.
- **Biology (Plants/Animals):** taxonomic_info, morphology, habitat, ecology, behavior, relationships.
- **Cultural Artifacts (Clothing/Tools/Objects):** physical_properties, construction_method, cultural_significance, functional_purpose, historical_period, regional_style, relationships.
- **Architecture/Structures:** structural_elements, style_period, dimensions, construction_details, cultural_context, relationships.
- **Technology/Obsolete Inventions:** physical_design, functional_mechanism, purpose, historical_context, relationships.

Output Format:
Return a JSON structure adapted to the domain, using the following core representation:

```
{
  "name": "<Entity name>",
  "domain": "<Domain category>",
  "type": "<Specific type within domain>",
  "origin": "<Geographic/cultural/historical origin>",
  "primary_sources": ["<Source 1>", "..."],
  "alternative_names": ["<Alternative name 1>", "..."],
  "visual_attributes": {
    // morphology, appearance, physical properties
  },
  "functional_attributes": {
    // abilities, behavior, cultural significance
  },
  "relational_attributes": {
    // relationships, pollinators, influenced_by
  },
  "contextual_attributes": {
    // habitat, historical period, geographic context
  }
}
```

Note: Adapt the JSON structure to include only fields relevant to the domain. Focus on attributes that support visual generation and contextual understanding.

Source Raw Text
`{raw_text}`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: RAVEL's contributions are clearly presented in the abstract and introduction, including knowledge graph-driven generation, optional SRD refinement, and context-aware editing. The claims are supported by quantitative experiments, ablations, qualitative comparisons, and human evaluation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in clearly. These include dependence on knowledge graph coverage and quality, additional latency from iterative self-correction (SRD), and challenges in scaling to large or noisy knowledge graphs. Potential bias inherited from source data is also acknowledged.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results or proofs; it is primarily an empirical and systems-oriented framework.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All components of the pipeline are described, including KG construction, retrieval, prompt synthesis, and SRD refinement. Experimental settings, datasets, and evaluation protocols are detailed in Appendix to enable reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides a public project repository and describes the benchmark suite, annotation schema, knowledge-graph construction, and evaluation procedures. Appendix F.2 specifies the benchmark and evaluation assets prepared for public release.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the evaluated image generators, comparison methods, benchmark composition, inference environment, and evaluation metrics. Section 3 and Appendices E and H document the retrieval procedure, prompt conditioning, SRD threshold, and refinement budget.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Table 1 reports mean scores and standard deviations across the primary generation metrics. Appendix E additionally reports mean attribute-retention, regression, and recovery rates with sample standard deviations across the five evaluated generators.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix A describe the single NVIDIA A100 inference environment, hosted model APIs, and computational workflow. Table 4 reports backbone-specific inference times, including the additional cost of iterative self-correction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly documented concept information, established generative models, and structured evaluation procedures. The human preference study uses informed consent and does not collect sensitive personal information. Responsible-use considerations are discussed in Appendix G.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix G discusses applications in education, scientific communication, and cultural representation, alongside transparency, knowledge-source reliability, cultural interpretation, and responsible use.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: RAVEL operates through external knowledge retrieval and text conditioning without releasing new pretrained image-generator weights. Appendix G discusses responsible use, source validation, and considerations for culturally sensitive applications.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (e.g., Wikipedia, Project Gutenberg) and models are properly cited. Licensing terms are respected, and new benchmarks will be released under CC BY 4.0.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce three new benchmarks (MythoBench, Rare-Concept-1K, Novel-Bench) and a structured KG pipeline. These are documented and will be released with full metadata and annotations.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: A user study was conducted using Prolific with 60 participants. Details of the task design, evaluation criteria, and participant instructions are provided in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The study involved minimal-risk evaluation of generated images and did not collect sensitive personal data. Formal IRB approval was not obtained, informed consent was obtained from all participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of the method, used for knowledge graph construction, query generation, prompt synthesis, and evaluation (LLM-as-a-judge). Their role is described in detail in Appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.