# ENABLING VISUAL REASONING OVER GENERATED IMAGES IN DIFFUSION MODELS

**Anonymous authors**
Paper under double-blind review

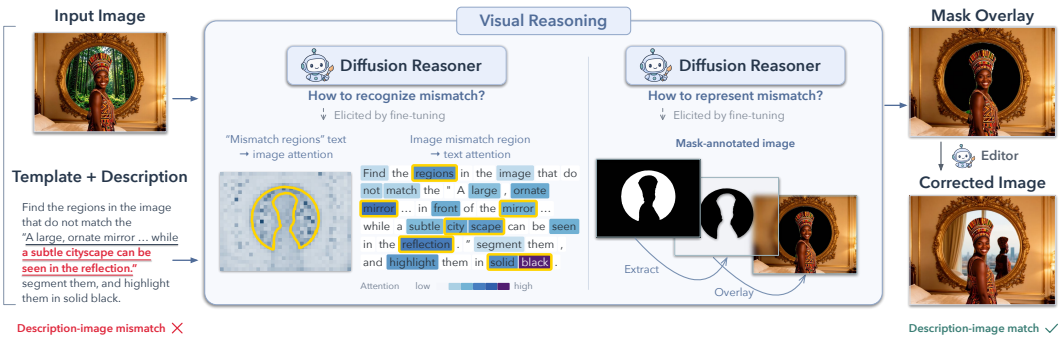



Figure 1: **Visual Reasoning in Diffusion Models.** Inspired by reasoning in LLMs, we enable diffusion models to generate explicit visual thoughts that autonomously identify mismatches, mark them with masks, and guide an SFT-adapted editor to correct the image.

## ABSTRACT

Reasoning has become central to LLMs, whereas diffusion models have largely been treated as visual generators. As image generation increasingly requires understanding complex and compositional prompts, existing diffusion models often rely on external LLMs or VLMs for reasoning, leaving instruction understanding decoupled from visual generation. We introduce `Seenk`, a visual reasoning paradigm for correcting mismatches between generated images and their text prompts. Given an image-description pair, `Seenk` determines whether any mismatches exist and, if so, generates a mask-annotated image to localize them. This image serves as an explicit visual thought, guiding an SFT-adapted editor to correct the identified mismatches. To equip diffusion models with this capability, we introduce two core visual reasoning modules: VR-Base and VR-Refiner. VR-Base is trained via SFT on `VRB-Dataset`, a visual reasoning dataset constructed through our data pipeline, to learn mismatch detection and visualization. VR-Refiner is trained via DPO on `VRR-Dataset`, a preference dataset constructed with a VLM, to further improve visual reasoning and mask localization accuracy. Experiments on visual reasoning and correction demonstrate that `Seenk` achieves competitive performance against strong open- and closed-source models, as well as pipelines that employ an LLM as an external reasoner. Further experiments show that, like LLMs, `Seenk` benefits from larger reasoning budgets, generating more visual thoughts to better handle complex prompts.

## 1 INTRODUCTION

Reasoning is a central capability that enables large language models (LLMs) to solve complex problems through intermediate steps (Wei et al., 2022). We view reasoning as a higher-order capability that builds on both understanding and generation. Current image generation systems must follow complex instructions, which requires reasoning beyond the denoising and reconstruction objectives of diffusion models. Reasoning in visual generative models, however, remains comparatively underexplored. LLMs show that reasoning can be developed on top of generative pretraining. This motivates a similar question in vision: *can visual generators develop reasoning from their own understanding and generation capabilities?*



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

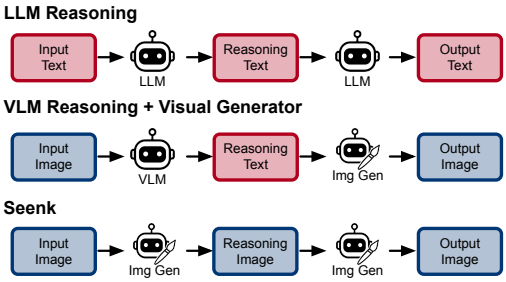



Figure 2: **Reasoning comparison.** LLMs reason through textual thoughts, while image generation often relies on external VLMs for reasoning. Inspired by LLMs, `Seenk` uses the visual generator's own outputs as intermediate visual thoughts to guide subsequent generation.

Several studies have used the reasoning capabilities of external LLMs or VLMs to guide visual generation. These models provide semantic representations, textual plans, or feedback (Mi et al., 2025; Yin et al., 2025; Li et al., 2025). The external model performs the reasoning, while the image generator follows its instructions. The generator thus relies on externally supplied reasoning rather than developing its own reasoning capability grounded in visual understanding and generation. Recent work, however, shows that fine-tuning can elicit visual understanding from pretrained visual generators, providing a potential foundation for developing their own reasoning capabilities. Vision Banana and SenseNova-Vision produce outputs such as segmentation masks and depth maps (Gabeur et al., 2026; Han et al., 2026). Together with surface normals and optical flow, these visual outputs capture spatial, geometric, and motion information that support visual judgment. This suggests a different path to reasoning: a visual generator can use its own visual outputs as intermediate thoughts to inspect and guide subsequent generation, much like LLMs use intermediate reasoning to produce final answers.

We study visual reasoning through a focused task: *identifying and correcting mismatches between an image and its description*. In practice, users typically need to generate multiple candidates to obtain an image that matches their description. This motivates the development of models that can inspect and revise an existing image. We introduce `Seenk`, which performs this process through explicit visual thoughts (Fig. 1). Given an image and its description, a diffusion reasoner identifies mismatched regions and marks them in black while preserving the rest of the image. The resulting mask annotated image serves as a visual thought that guides an editor to correct the mismatches. As shown in Fig. 2, these visual thoughts are generated by the visual generator itself, analogous to the intermediate reasoning produced by LLMs before a final answer.

To equip diffusion models with visual reasoning, we introduce two core modules, VR-Base and VR-Refiner, which are trained using a two-stage strategy inspired by the SFT and optimization paradigm in LLMs (Guo et al., 2025a). We first construct visual reasoning data and use it to train VR-Base through SFT, teaching a pretrained image generator to inspect an image against a target description and mark mismatched regions with masks. However, SFT often produces imperfect masks, including near-black pixels, inaccurate boundaries, and stray pixels outside the target regions. We therefore train VR-Refiner through direct preference optimization (DPO) (Wallace et al., 2024) with VLM-based preferences to further improve visual reasoning and produce more accurate, cleaner masks. An SFT-adapted editor then corrects the masked regions according to the description.

We design two scalable data pipelines to construct `VRB-Dataset` and `VRR-Dataset` for training VR-Base and VR-Refiner, respectively. For `VRB-Dataset`, we start from image description pairs in an existing text to image dataset (Ma et al., 2026). A VLM guides image editing to create and verify diverse mismatches, then uses grounding and segmentation to locate the affected regions. We overlay these regions in black to form visual thought targets, paired with the description, inconsistent image, and original image. For `VRR-Dataset`, we sample visual thoughts with the VR-Base and use a VLM to rank them by mismatch coverage and false positive regions. The resulting preference pairs are used to train the VR-Refiner.

Experiments show that `Seenk` localizes image description mismatches more accurately than the evaluated baselines. It achieves a mask IoU of 0.775, compared with 0.356 for the strongest baseline. These visual thoughts also improve mismatch correction. Our method reaches a VQAScore of 0.752, exceeding the best open source baseline at 0.709, while maintaining competitive visual fidelity. On cases with two mismatches, successive or complementary visual thoughts can further improve correction. On MagicBrush (Zhang et al., 2023), `Seenk` achieves the best DINO and CLIP-I scores among the compared methods, showing strong cross-dataset generalization. Overall, these results show that explicit visual thoughts help visual generators inspect and improve their own outputs.

# 2 Related Works

## 2.1 Reasoning in LLMs

Reasoning in LLMs has developed along several lines. Chain-of-thought prompting and self-consistency elicit explicit intermediate steps (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022). Rationale bootstrapping, process supervision, and preference optimization further improve reasoning through training (Zelikman et al., 2022; Lightman et al., 2024; Pang et al., 2024). Search and interaction extend linear reasoning into exploration and action (Yao et al., 2023; 2022). More recent work improves reasoning with reinforcement learning and additional test-time computation (Jaech et al., 2024; Guo et al., 2025a; Snell et al., 2024). In contrast, diffusion models still lack reasoning mechanisms expressed through explicit visual intermediate states.

## 2.2 Reasoning for Image Generation

Generative visual training can provide a strong foundation for visual understanding. Vision Banana and SenseNova-Vision formulate diverse perception tasks as image or multimodal generation (Gabeur et al., 2026; Han et al., 2026). However, understanding alone does not enable a model to explicitly inspect a generated result and decide whether further action is needed.

### 2.2.1 Reasoning in Unified Multimodal Models for Image Generation

Unified multimodal models combine visual understanding and generation within a shared framework (Team, 2024; Chen et al., 2025b; Xie et al., 2026; Chen et al., 2025a; Deng et al., 2025; OpenSenseNova, 2026). Their language knowledge and semantic representations can improve visual generation (Sun et al., 2024; Tong et al., 2025; Shi et al., 2026). Recent methods make this interaction more explicit by interleaving textual reasoning with image generation, sometimes with visual subgoals (Huang et al., 2025; Wang et al., 2025; Chern et al., 2025; Guo et al., 2025b; Duan et al., 2026). However, planning, critique, and control remain mainly text driven. Our work instead studies reasoning expressed directly through spatial visual states.

### 2.2.2 Reasoning in Diffusion Models for Image Generation

Diffusion-based reasoning methods mainly follow three directions. One line uses external LLMs or VLMs to provide semantic representations, plans, or feedback for diffusion models (Koh et al., 2023; Zheng et al., 2026; Han et al., 2025; Mi et al., 2025; Yin et al., 2025; Li et al., 2025). Another improves inference through sampling, search, steering, or latent refinement (Singhal et al., 2025; Zhang et al., 2025; Jang et al., 2026; Wang et al., 2026). These methods add reasoning-like computation but do not produce explicit reasoning states. CoF-T2I instead treats intermediate video frames as visual reasoning, but focuses on a fixed semantics-to-aesthetics process (Tong et al., 2026). Our method differs by letting a diffusion model identify image-text mismatches and express them as spatially explicit visual thoughts for correction.

# 3 Method

While reasoning has advanced LLMs, reasoning-enhanced diffusion systems often rely on external VLMs for textual deliberation. We introduce a visual reasoning paradigm in which diffusion models express intermediate thoughts through image generation itself. Section 3.1 reviews the flow-matching and Diffusion-DPO objectives. Section 3.2 formulates the task and describes training VR-Base with SFT and VR-Refiner with DPO, together with SFT adaptation of an image editor to execute visual thoughts (Fig. 3). Finally, Section 3.3 details the construction of `VRB-Dataset` and `VRR-Dataset`, together with the evaluation benchmark derived from `VRB-Dataset`.

## 3.1 Preliminary

**Flow Matching.** Flow matching (Lipman et al., 2022; Liu et al., 2022) learns a velocity field that transports samples between noise and data distributions. Let $x_0$ denote the latent representation of a

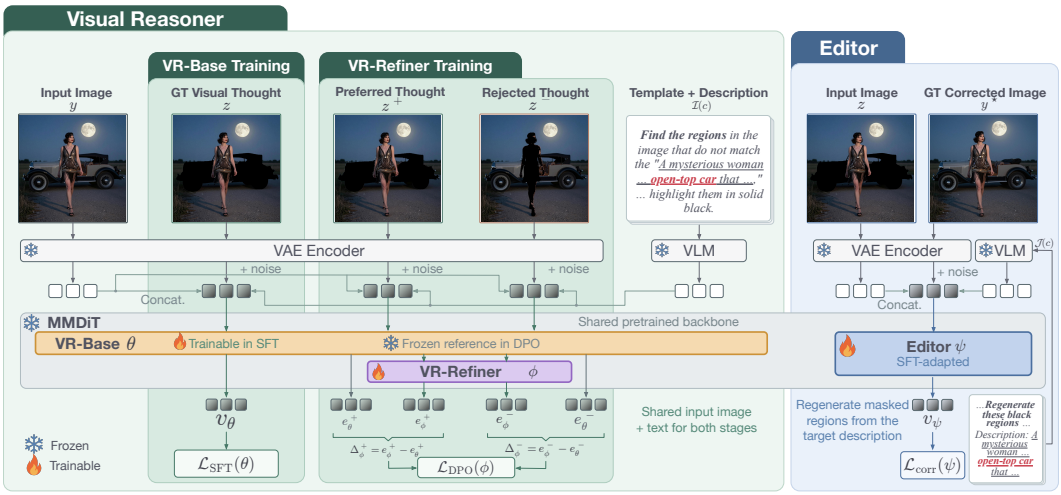



Figure 3: **Training pipeline.** VR-Base $\theta$ is first trained with flow-matching SFT to identify image-text inconsistencies and generate mask-annotated visual thoughts. VR-Refiner $\phi$ then refines these spatial judgments through DPO on preferred and rejected thoughts, using the frozen VR-Base as the reference. The editor is separately adapted through SFT to execute the visual thoughts and regenerate marked regions according to the target description.

target image. With $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim p(t)$ on $[0, 1]$, the noisy target and velocity are

$$x_t = (1 - t)x_0 + t\epsilon, \qquad u_t = \epsilon - x_0. \tag{1}$$

Conditioned on an image $a$ and a text prompt $b$, the velocity predictor is trained through

$$\ell_\theta(x_0, a, b; t, \epsilon) = \|v_\theta(x_t, t, a, b) - u_t\|_2^2, \quad \mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(x_0,a,b),\, t,\, \epsilon}\left[\ell_\theta(x_0, a, b; t, \epsilon)\right]. \tag{2}$$

**Diffusion DPO.** Diffusion-DPO (Wallace et al., 2024) learns from pairwise preferences by contrasting prediction errors against a frozen reference model. We apply its objective to flow-matching velocity predictions. Given preferred and rejected target latents $(x_0^+, x_0^-)$ conditioned on $(a, b)$, both candidates share the sampled timestep $t$ and noise $\epsilon$. For a trainable predictor $v_\phi$, define

$$\Delta_\phi^s = \ell_\phi(x_0^s, a, b; t, \epsilon) - \ell_{\mathrm{ref}}(x_0^s, a, b; t, \epsilon), \qquad s \in \{+, -\}, \tag{3}$$

where $\ell_{\mathrm{ref}}$ is computed using the frozen reference. The preference objective is

$$\mathcal{L}_{\mathrm{DPO}}(\phi) = -\mathbb{E}\left[\log \sigma\left(-\frac{\beta}{2}\left(\Delta_\phi^+ - \Delta_\phi^-\right)\right)\right], \tag{4}$$

where the expectation is over preference pairs, timesteps, and noise, $\sigma$ is the sigmoid function, and $\beta > 0$ controls the preference scale. This objective favors the preferred output through its relative prediction error, without fitting a separate reward model.

### 3.2 Enabling Visual Reasoning in Diffusion Models

#### 3.2.1 Motivation and Problem Formulation

Inspired by explicit reasoning in LLMs (Wei et al., 2022; Guo et al., 2025a) and visual understanding in pretrained generative models (Gabeur et al., 2026; Han et al., 2026), we study whether diffusion models can generate explicit visual thoughts for image correction. Given an image and description $c$, the model detects and localizes image-text inconsistencies without external cues. We denote an inconsistent image as $y$, its correct counterpart as $y^\star$, and the mask-annotated visual thought as $z$. The thought guides an editor to correct $y$, while a consistent image is left unchanged.

### 3.2.2 Eliciting Visual Reasoning through Supervised Fine-Tuning

We represent visual thoughts as RGB images with solid-black masks (Kirillov et al., 2023). Given an inconsistent image $y$ and binary mask $m$, where $m_{ij} = 1$ marks a region to correct, the target is

$$z = (1 - m) \odot y + m \odot \mathbf{0}, \tag{5}$$

where $\mathbf{0}$ denotes black and $m$ is broadcast across RGB channels. This preserves the surrounding image while providing a clear spatial cue for editing. For a correct image $y^\star$, we set $z = y^\star$.

We train VR-Base by fine-tuning Qwen-Image-Edit-2511 (Wu et al., 2025) on `VRB-Dataset`, optimizing parameters $\theta$. Each example $(a, c, z)$ uses $a = y$ for inconsistent images and $a = y^\star$ otherwise. The prompt $\mathcal{I}(c)$ combines a fixed visual inspection instruction with the description $c$. With $x_0 = \mathrm{Enc}(z)$ and the noisy target defined by Eq. 1, the SFT objective is

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(a,c,z),\, t,\, \epsilon} \left[ \ell_\theta \left( \mathrm{Enc}(z), a, \mathcal{I}(c); t, \epsilon \right) \right], \tag{6}$$

where Enc is the VAE encoder and $\ell_\theta$ is defined in Eq. 2. This trains the model to mark inconsistencies with visual thoughts and leave correct images unchanged.

### 3.2.3 Refining Visual Reasoning through Direct Preference Optimization

VR-Base in Sec. 3.2.2 elicits visual reasoning and teaches the model to express visual thoughts as mask-annotated images. However, the masks may contain near-black pixels, inaccurate boundaries, and stray black pixels, as shown in App. E. Inspired by reinforcement learning for LLM reasoning (Guo et al., 2025a), we train VR-Refiner through preference optimization to refine visual thoughts. We use Diffusion-DPO from Sec. 3.1 with spatially weighted prediction errors.

For each inconsistent image $y$ and description $c$, `VRR-Dataset` provides a preferred thought $z^+$ and a rejected thought $z^-$. We initialize $v_\phi$ from VR-Base and keep $v_\theta$ frozen as the reference. Only the parameter $\phi$ of VR-Refiner is optimized. For $s \in \{+, -\}$, we set $x_0^s = \mathrm{Enc}(z^s)$ and construct $x_t^s$ and $u_t^s$ using Eq. 1, with shared $t$ and $\epsilon$ across the pair.

Since mask regions are often small, uniform error averaging can weaken the preference signal. We therefore assign larger weights to regions where the two candidates disagree. Let $M^+$ and $M^-$ denote their pure-black mask regions, excluding pixels already black in $y$. We define

$$w(q) = \begin{cases} \lambda_1, & q \in (M^+ \cup M^-) \setminus (M^+ \cap M^-), \\ \lambda_2, & q \in M^+ \cap M^-, \\ \lambda_3, & \text{otherwise.} \end{cases} \tag{7}$$

Here, $\lambda_1 > \lambda_2 > \lambda_3 > 0$ are fixed hyperparameters that prioritize mask disagreements while retaining supervision over shared mask regions and the surrounding context.

Using the resulting weight map $W$, we define the policy's weighted prediction error as

$$e_\phi^s = \frac{\sum_p W_p \left\| v_\phi(x_t^s, t, y, \mathcal{I}(c))_p - u_{t,p}^s \right\|_2^2}{\sum_p W_p}, \tag{8}$$

where $p$ indexes latent spatial locations. The reference error $e_\theta^s$ is computed analogously, using the same weight map for both candidates and both models. We then set

$$\Delta_\phi^s = e_\phi^s - e_\theta^s \tag{9}$$

and optimize $\phi$ with Eq. 4. Constant channel normalization is absorbed into $\beta$. This objective increases the preferred candidate's relative advantage in weighted prediction error over the rejected alternative, measured against the frozen SFT reference.

### 3.2.4 Executing Visual Thoughts for Correction

We separately fine-tune Qwen-Image-Edit-2511 on `VRB-Dataset` as an editor with parameters $\psi$. Given a visual thought $z$ and description $c$, the editor regenerates the marked regions while preserving the rest of the image. Each example is $(z, c, y^\star)$. Let $\mathcal{J}(c)$ denote the text prompt formed by combining a fixed image correction instruction with the description $c$. With $x_0 = \mathrm{Enc}(y^\star)$ and the noisy target defined by Eq. 1, the correction objective is

$$\mathcal{L}_{\mathrm{corr}}(\psi) = \mathbb{E}_{(z,c,y^\star),\, t,\, \epsilon} \left[ \ell_\psi \left( \mathrm{Enc}(y^\star), z, \mathcal{J}(c); t, \epsilon \right) \right]. \tag{10}$$

### 3.3 DATA CONSTRUCTION

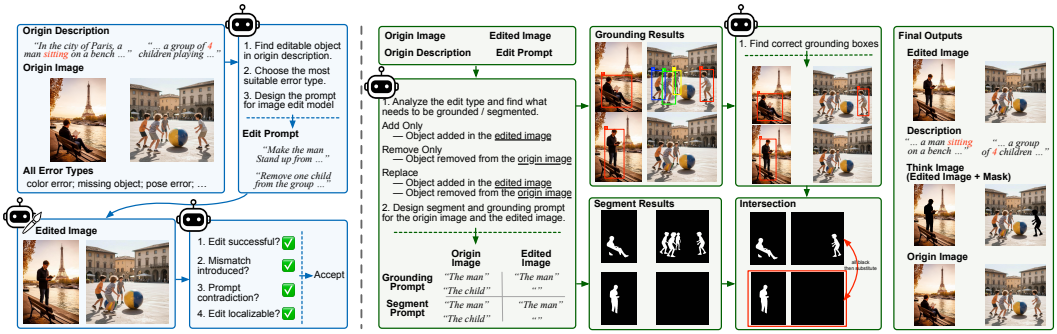



Figure 4: **Construction pipeline of `VRB-Dataset`.** Starting from a description $c$ and a corresponding correct image $y^\star$, an MLLM designs an edit that introduces an inconsistency, producing $y$ through an image editor. The MLLM then guides object grounding, candidate selection, and segmentation to obtain the intervention annotation $m$. Overlaying $m$ in solid black on $y$ yields the visual thought $z$, while $y^\star$ provides the correct reference.

#### 3.3.1 SFT DATASET CONSTRUCTION

As shown in Fig. 4, we construct `VRB-Dataset` from Fine-T2I image-description pairs $(y^\star, c)$ (Ma et al., 2026). For each pair, Qwen3-VL-235B-A22B-Thinking selects a target region and one of 24 error types, such as color, object, or counting errors. It then generates an edit instruction that conflicts with an explicit detail in $c$. An image editor applies the instruction to the original correct image $y^\star$ to produce an inconsistent image $y$. The MLLM verifies the intended mismatch, edit locality, and image quality, retaining only successful edits for annotation.

To construct the visual thought, the MLLM identifies the affected objects and provides prompts for Grounding DINO (Liu et al., 2023) and SAM 3 (Carion et al., 2025). The MLLM then selects the grounding proposals corresponding to the affected objects using the description and edit instruction, constraining segmentation to the intended regions. We construct the visual thought $z$ by covering the annotated regions of $y$ with solid-black masks. If the annotation on $y$ is empty, we instead overlay the annotation from $y^\star$ onto $y$, marking the original locations of objects removed from the correct image during error synthesis. The selected binary annotation serves as $m$ in Eq. 5. We organize the data as quadruplets $(c, y, z, y^\star)$, comprising the text description, the edited image containing inconsistencies, the visual thought, and the original correct image, respectively.

The construction pipeline yields 48,421 valid records, of which 47,421 are used for training. From the remaining 1,000 held-out records, we curate a 100-example evaluation benchmark. Our selection prioritizes broad coverage across error categories and high-quality mask annotations, providing diverse test cases for evaluating visual reasoning capabilities.

#### 3.3.2 DPO DATASET CONSTRUCTION

We use the first 10,000 training samples from `VRB-Dataset` to construct `VRR-Dataset`. For each $(y, c)$, we sample eight outputs from the SFT reasoner under $\mathcal{I}(c)$ with different seeds. We then extract each predicted mask and overlay it in solid black on $y$. This restores the input pixels outside the mask, removing unintended changes that may arise during visual thought generation.

We use Qwen3.5-9B as a lightweight VLM judge to compare the eight candidates in a single evaluation. The judge receives $c$, $y$, the correct reference $y^\star$, and structured edit information, including available localization hints. Candidate masks are displayed as translucent overlays to keep the underlying content visible. The judge evaluates coverage of inconsistent content, avoidance of correct regions, and spatial precision, selecting the best and worst candidates as $z^+$ and $z^-$. We discard invalid judgments and pairs with negligible mask coverage or insufficient spatial disagreement, yielding `VRR-Dataset` with 8,539 preference pairs of the form $(y, c, z^+, z^-)$. These pairs provide VLM-derived preference supervision that encourages more accurate spatial judgments, while the standardized targets reinforce the prescribed mask-annotated visual thought format.

Table 1: **Quantitative evaluation of mismatch localization.** Text- and image-based reasoning are compared using pixel, grid, and bounding-box IoU. `Seenk` w/o VR-Refiner uses VR-Base alone.

| Reasoning Modal | Method | IoU↑ | Grid IoU↑ (64 × 64) | Grid IoU↑ (8 × 8) | BBox IoU↑ |
|---|---|---|---|---|---|
| Text | Bagel | 0.218 | 0.242 | 0.337 | 0.286 |
| | SenseNova-U1 | 0.080 | 0.096 | 0.194 | 0.171 |
| | Qwen3-VL | 0.356 | 0.382 | 0.474 | 0.384 |
| Image | GPT-Image-2 | 0.087 | 0.134 | 0.248 | 0.210 |
| | Nano Banana Pro | 0.225 | 0.316 | 0.408 | 0.369 |
| | FLUX.2-dev | 0.199 | 0.245 | 0.343 | 0.288 |
| | Qwen-Image-Edit | 0.167 | 0.225 | 0.339 | 0.272 |
| | HunyuanImage | 0.051 | 0.118 | 0.279 | 0.233 |
| | Bagel | 0.012 | 0.039 | 0.138 | 0.129 |
| | SenseNova-U1 | 0.005 | 0.061 | 0.219 | 0.226 |
| | `Seenk` w/o VR-Refiner | 0.768 | 0.810 | 0.807 | 0.626 |
| | `Seenk` | **0.775** | **0.814** | **0.810** | **0.627** |

# 4 EXPERIMENTS

We evaluate visual reasoning from four perspectives. Sec. 4.2 measures mismatch localization accuracy, and Sec. 4.3 evaluates how visual thoughts improve correction. Sec. 4.4 studies the effect of additional computation on complex cases. Finally, Sec. 4.5 evaluates generalization to unseen data distributions. Experimental setup is provided in Sec. 4.1.

## 4.1 EXPERIMENTAL SETUP

**Training.** We train Qwen-Image-Edit-2511 for visual reasoning with SFT followed by DPO on the datasets in Sec. 3.3. For VR-Base, we use paired edited and reasoning images from `VRB-Dataset`, with LoRA rank 64, a learning rate of $5 \times 10^{-5}$, and 30k training steps on 96 NVIDIA H20 GPUs. Building on VR-Base, we train VR-Refiner on `VRR-Dataset` with LoRA rank 16, a learning rate of $1 \times 10^{-5}$, and 28k steps on 32 NVIDIA H20 GPUs. Both stages use AdamW. To accommodate the solid-black masked regions in reasoning images, we also adapt the Qwen-Image-Edit-2511 executor using paired reasoning and original images from `VRB-Dataset`, with LoRA rank 64 and AdamW for 50k steps on 96 NVIDIA H20 GPUs at a learning rate of $1 \times 10^{-5}$.

**Metrics.** We evaluate mismatch localization and correction quality. For localization, Pixel IoU measures mask overlap with ground truth. Grid IoU at $8 \times 8$ and $64 \times 64$ measures spatial coverage, while BBox IoU compares the tight bounding boxes of the masks. Masks are extracted from black regions in reasoning images or rasterized from MLLM predictions, excluding existing black pixels. For correction, VQAScore and CLIP-T measure image text alignment, while DINO and CLIP-I measure similarity to the ground truth image. See section A for details.

## 4.2 SPATIAL PRECISION OF VISUAL REASONING

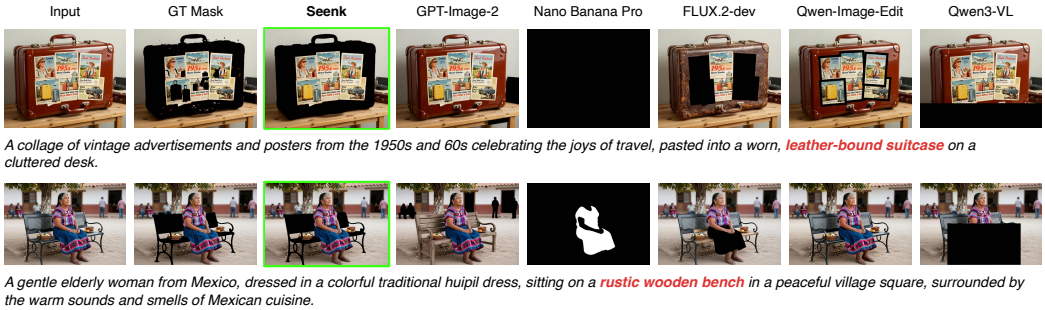



Figure 5: **Comparison of predicted mismatch regions.** See App. H for all baselines.

Table 2: **Mismatch correction across reasoning protocols.** `Seenk` variants progressively introduce VR-Base, SFT fine-tuned editor, and VR-Refiner (Full).

| Method | Reasoning | VQAScore↑ | CLIP-T↑ | DINO↑ | CLIP-I↑ |
|---|---|---|---|---|---|
| GPT-Image-2 | Unknown | 0.803 | 0.308 | 0.878 | 0.914 |
| Nano Banana Pro | Unknown | 0.726 | 0.305 | 0.919 | 0.938 |
| Bagel | Implicit | 0.553 | 0.298 | 0.837 | 0.894 |
| SenseNova-U1 | Implicit | 0.709 | 0.305 | 0.866 | 0.908 |
| FLUX.2-dev | Implicit | 0.705 | 0.307 | **0.895** | 0.923 |
| Qwen-Image-Edit | Implicit | 0.679 | 0.307 | 0.866 | 0.909 |
| HunyuanImage | Implicit | 0.587 | 0.301 | 0.884 | 0.923 |
| Bagel | Explicit, Txt. | 0.490 | 0.285 | 0.716 | 0.794 |
| SenseNova-U1 | Explicit, Txt. | 0.698 | 0.307 | 0.815 | 0.885 |
| MLLM+Edit Model | Explicit, Txt. | 0.652 | 0.303 | 0.875 | 0.906 |
| ReasonEdit | Explicit, Txt. | 0.510 | 0.293 | 0.872 | 0.910 |
| Bagel | Explicit, Img. | 0.413 | 0.288 | 0.800 | 0.836 |
| SenseNova-U1 | Explicit, Img. | 0.632 | 0.306 | 0.880 | 0.883 |
| `Seenk` (1. VR-Base) | Explicit, Img. | 0.723 | 0.307 | 0.877 | 0.914 |
| `Seenk` (1. VR-Base, 2. SFT) | Explicit, Img. | 0.750 | 0.307 | 0.891 | 0.924 |
| `Seenk` (Full) | Explicit, Img. | **0.752** | **0.308** | 0.890 | **0.925** |

**Baselines.** We use tailored prompt templates to instruct baselines to localize image-description mismatches in two forms: bounding-box coordinates for text-based methods and solid-black masks over the input image for image-based methods, enabling consistent localization evaluation.

**Results Analysis.** Tab. 1 and Fig. 5 show that `Seenk` achieves the best performance across all four metrics. Text-based baselines struggle with fine-grained localization, while several image-based baselines produce black masks from instructions but misidentify the mismatch regions. Comparing Qwen-Image-Edit, `Seenk` w/o VR-Refiner, and `Seenk` highlights the contributions of both training stages: VR-Base raises IoU from 0.167 to 0.768, while VR-Refiner further improves it to 0.775, providing modest but consistent gains across all metrics.

### 4.3 Visual Reasoning for Mismatch Correction

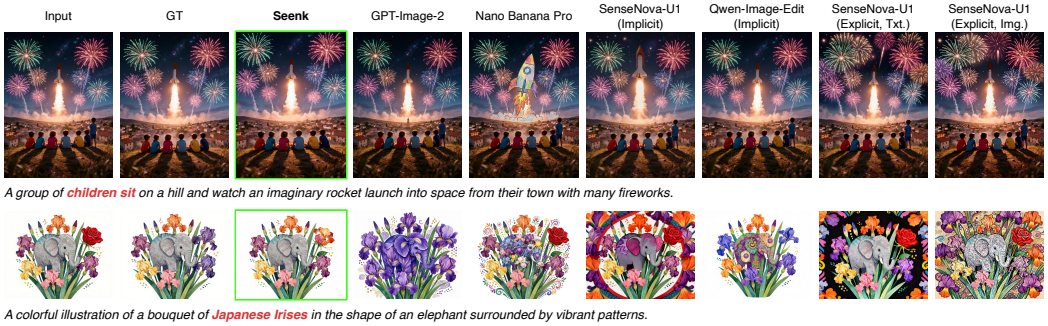



Figure 6: **Qualitative comparisons.** See App. H for all baselines.

**Baselines.** We compare four groups: (i) proprietary reference models, ChatGPT-Image-2 and Nano Banana Pro, whose reasoning procedures are undisclosed; (ii) direct-editing models prompted to correct mismatches without explicit intermediate outputs; (iii) explicit text-based methods, using either ReasonEdit's native thinking and reflection or bounding-box localization followed by correction; and (iv) explicit image-based methods that generate mask-annotated images before correction. For bounding-box methods, predicted coordinates are rendered as black overlays for the editor.

**Results Analysis.** Tab. 2 and Fig. 6 show that `Seenk` achieves the best VQAScore, CLIP-T, and CLIP-I among open-source methods, and ranks third in DINO. It matches GPT-Image-2 in CLIP-T and surpasses it in DINO and CLIP-I. Adding VR-Base to the untuned Qwen-Image-Edit improves VQAScore from 0.679 to 0.723. Fine-tuning the editor further raises it to 0.750, while VR-Refiner improves it to 0.752. DINO and CLIP-I also improve across these stages.

Table 3: **Cross-dataset generalization on MagicBrush.**

| Method | Reasoning | CLIP-T↑ | DINO↑ | CLIP-I↑ |
|---|---|---|---|---|
| FLUX.2-dev | Implicit | 0.271 | 0.797 | 0.886 |
| Qwen-Image-Edit | Implicit | 0.272 | 0.787 | 0.883 |
| SenseNova-U1 | Explicit, Txt. | 0.267 | 0.630 | 0.811 |
| MLLM+Edit Model | Explicit, Txt. | **0.276** | 0.751 | 0.862 |
| Bagel | Explicit, Img. | 0.265 | 0.732 | 0.836 |
| SenseNova-U1 | Explicit, Img. | 0.255 | 0.645 | 0.782 |
| Seenk | Explicit, Img. | 0.272 | **0.798** | **0.888** |

### 4.4 Reasoning Effort and Problem Complexity

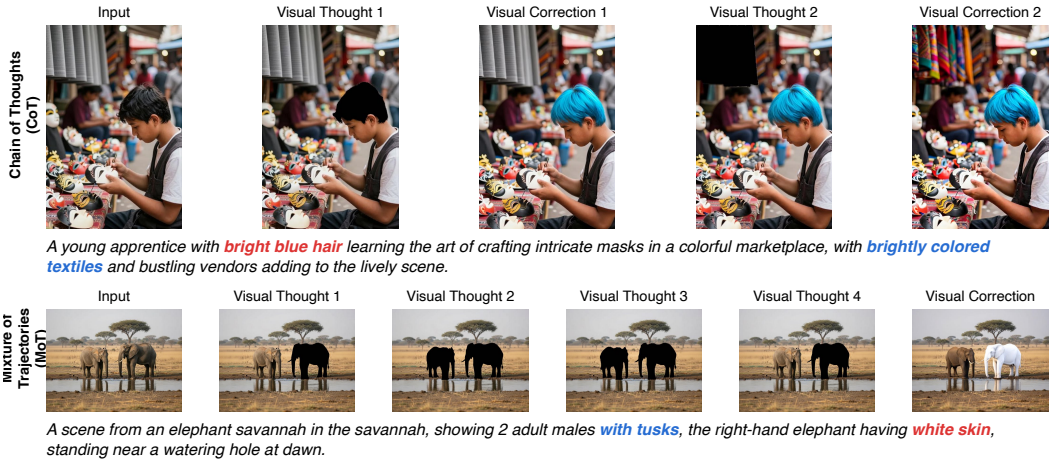



Figure 7: **More visual reasoning for multiple mismatches.**

Motivated by extended reasoning in LLMs, we increase visual reasoning effort for images with two mismatches. Chain of Thoughts (CoT) alternates reasoning and editing to correct errors sequentially. Mixture of Trajectories (MoT) unions masks from four sampled visual thoughts for joint correction. In Fig. 7, CoT corrects the apprentice's hair and textile colors in sequence, while MoT combines complementary detections to correct the elephant's skin color and missing tusk. These examples suggest that additional visual reasoning can help resolve multiple errors. A broader study of reasoning budgets and more complex cases is left for future work.

### 4.5 Generalization to Unseen Data Distributions

We evaluate generalization on 1,053 MagicBrush editing turns (Zhang et al., 2023). Each input combines the source image, original description, and edit instruction, with the edited image as ground truth. We compare two baselines from each reasoning category in Table 2. We omit VQAScore because the combined text does not always describe the target image, and even ground truth scores only 0.216. CLIP-T is retained as a supplementary metric for text-image alignment. As shown in Table 3, Seenk achieves the best DINO (0.798) and CLIP-I (0.888), with a competitive CLIP-T of 0.272. These results show strong generalization across data distributions.

## 5 Conclusion

We introduced Seenk, a visual reasoning paradigm for diffusion models inspired by explicit reasoning in LLMs. The reasoner identifies image text mismatches and expresses them as mask annotated visual thoughts to guide correction. We train VR-Base with SFT and VR-Refiner with DPO to get the visual reasoner, and train an editor with SFT using automatically constructed datasets. Our current design uses separate LoRA adapters and requires additional inference. Future work could integrate visual reasoning into pretraining, support richer operations and representations such as super resolution and depth, and better use additional computation for complex visual reasoning.

### AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, and assisting with translation. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results are not applicable to this work. Additionally, we used generative AI tools for summarizing or analyzing existing literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

To support reproducibility, we document our method, experimental setup, and dataset construction in the main text and appendix. Sec. 3.2 specifies the visual thought representation and the training objectives for the reasoner and the SFT-adapted editor. Sec. 4.1 reports the backbone, optimization settings, training budgets, and evaluation metrics, with metric calculation and DPO implementation details in App. A and C. App. D provides the prompt templates used by our method and the baselines. For dataset reconstruction, Sec. 3.3 describes error synthesis, mask annotation, and preference-pair construction, while App. G specifies error categories, data splits, pair-filtering criteria, and dataset statistics.

## REFERENCES

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

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*, 2025b.

Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv preprint arXiv:2505.22525*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of autoregressive visual generation with reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 118771–118790, 2026.

Valentin Gabeur, Shangbang Long, Songyou Peng, Paul Voigtlaender, Shuyang Sun, Yanan Bao, Karen Truong, Zhicheng Wang, Wenlei Zhou, Jonathan T Barron, et al. Image generators are generalist vision learners. *arXiv preprint arXiv:2604.20329*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, et al. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025b.

Feng Han, Yang Jiao, Shaoxiang Chen, Junhao Xu, Jingjing Chen, and Yu-Gang Jiang. Controlthinker: Unveiling latent semantics for controllable image generation through visual reasoning. *arXiv preprint arXiv:2506.03596*, 2025.

Xiaoyang Han, Jianhua Li, Kewang Deng, Zukai Chen, Xuanke Shi, Sihan Wang, Boxuan Li, Linyan Wang, Siyi Xie, Xin You, et al. Vision as unified multimodal generation. *arXiv preprint arXiv:2607.06560*, 2026.

Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, et al. Interleaving reasoning for better text-to-image generation. *arXiv preprint arXiv:2509.06945*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Saining Xie, Jaehong Yoon, and Sung Ju Hwang. Self-refining video sampling. *arXiv preprint arXiv:2601.18577*, 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36:21487–21506, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15657–15668, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xu Ma, Yitian Zhang, Qihua Dong, and Yun Fu. Fine-t2i: An open, large-scale, and diverse dataset for high-quality t2i fine-tuning, 2026. URL `https://arxiv.org/abs/2602.09439`.

Zhenxing Mi, Kuan-Chieh Wang, Guocheng Qian, Hanrong Ye, Runtao Liu, Sergey Tulyakov, Kfir Aberman, and Dan Xu. I think, therefore i diffuse: enabling multimodal in-context reasoning in diffusion models. *arXiv preprint arXiv:2502.10458*, 2025.

OpenSenseNova. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. https://github.com/OpenSenseNova/SenseNova-U1, 2026. GitHub repository.

Richard Y Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Lin, Luke Zettlemoyer, and Lili Yu. Lmfusion: Adapting pretrained language models for multimodal generation. *Advances in Neural Information Processing Systems*, 38:9163–9182, 2026.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14398–14409, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Chengzhuo Tong, Mingkun Chang, Shenglong Zhang, Yuran Wang, Cheng Liang, Zhizheng Zhao, Ruichuan An, Bohan Zeng, Yang Shi, Yifan Dai, et al. Cof-t2i: Video models as pure visual reasoners for text-to-image generation. *arXiv preprint arXiv:2601.10061*, 2026.

Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17001–17012, 2025.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, et al. Demystifing video reasoning. *arXiv preprint arXiv:2603.16870*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yi Wang, Mushui Liu, Wanggui He, Hanyang Yuan, Longxiang Zhang, Ziwei Huang, Guanghao Zhang, Wenkai Fang, Haoze Jiang, Shengxuming Zhang, et al. Towards enhanced image generation via multi-modal chain of thought in unified generative models. *arXiv preprint arXiv:2503.01298*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *Advances in Neural Information Processing Systems*, 38:47490–47518, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, et al. Reasonedit: Towards reasoning-enhanced image editing models. *arXiv preprint arXiv:2511.22625*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in neural information processing systems*, 36:31428–31449, 2023.

Tao Zhang, Jia-Shu Pan, Ruiqi Feng, and Tailin Wu. Vfscale: Intrinsic reasoning through verifier-free test-time scalable diffusion model. *arXiv preprint arXiv:2502.01989*, 2025.

Kaizhi Zheng, Xuehai He, and Xin Eric Wang. Interleaved vision-and-language generation via generative voken. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 472–482, 2026.

# A More details about Metric Calculation

A pixel is considered fully black only when all three decoded RGB channels equal zero. We exclude pixels already black in the corresponding input reference from both predicted and ground-truth masks, without applying a near-black tolerance. BBox IoU uses one axis-aligned bounding rectangle per mask, enclosing all disconnected regions. All IoU variants assign a score of 1 when both masks are empty and 0 when only one is empty, and scores are averaged across images.

For correction metrics, DINO uses `dinov3-vitl16-pretrain-lvd1689m`, while CLIP-I and CLIP-T use `clip-vit-large-patch14`. DINO and CLIP-I compute cosine similarity between corrected and ground-truth image embeddings; CLIP-T computes cosine similarity between corrected-image and target-description embeddings. VQAScore is computed through `t2v_metrics` using the `qwen2.5-vl-7b` configuration with the Qwen2.5-VL-7B-Instruct checkpoint.

# B Evaluation of the SFT Editor

We evaluate the SFT Editor for correction with provided masks, where each method receives a ground-truth mask-annotated image and the target description. As shown in Table 4, our SFT Editor achieves the highest VQAScore among the open-source methods (0.750), approaching GPT-Image-2 (0.751), and improves all four metrics over its Qwen-Image-Edit-2511 base model. These results demonstrate its ability to execute the provided spatial guidance and produce images that better match the target description.

Fig. 8 illustrates how the SFT Editor restores the specified sitting and walking poses while preserving the surrounding scene.

Table 4: **Correction with provided masks.** All methods use the same ground-truth mask annotations. Higher values are better for all metrics. The best result within each source category is shown in bold.

| Method | VQAScore↑ | CLIP-T↑ | DINO↑ | CLIP-I↑ |
|---|---|---|---|---|
| *Closed-source* | | | | |
| GPT-Image-2 | **0.751** | **0.306** | **0.911** | 0.930 |
| Nano Banana Pro | 0.735 | 0.305 | 0.904 | **0.932** |
| *Open-source* | | | | |
| FLUX.2-dev | 0.706 | **0.309** | **0.903** | **0.929** |
| Qwen-Image-Edit-2511 | 0.714 | 0.304 | 0.883 | 0.919 |
| **SFT Editor (Ours)** | **0.750** | 0.307 | 0.893 | 0.925 |

# C DPO Implementation Details

This section details the implementation of spatially weighted preference optimization and the configuration of the policy and reference models.

## C.1 Spatially Weighted Preference Optimization

**Spatial weights.** We set the fixed weights in Eq. 7 to

$$(\lambda_1, \lambda_2, \lambda_3) = (20, 2, 1), \tag{11}$$

corresponding to the symmetric difference, intersection, and background of the preferred and rejected masks, respectively. The largest weight emphasizes regions where the candidates disagree. The smaller weights retain supervision over shared mask regions and the surrounding image.

**Mapping weights to latent space.** We construct the three region labels from the pure-black pixels of the candidate images, excluding pixels already pure black in the input image. When resizing is




Figure 8: **Qualitative results of correction with provided masks.** All methods receive the same ground-truth mask-annotated image and target description. The three examples concern the sitting poses of a merchant and a woman on a couch, and a person's walking pose in snow. Bold red text highlights the description content relevant to each mismatch, and green outlines indicate our SFT Editor.

required, the discrete region labels are resized using nearest-neighbor interpolation. We then apply adaptive max pooling separately to the binary maps of the intersection and symmetric difference to match the latent spatial resolution. If both regions map to the same latent location, the symmetric difference takes priority and receives $\lambda_1$. Locations containing only intersection pixels receive $\lambda_2$, and all remaining locations receive $\lambda_3$. The resulting weight map is shared by both candidates and both the policy and reference models.

**Loss normalization and preference scale.** The implementation additionally averages the weighted prediction error over latent channels, computing $\bar{e}^s_\phi = e^s_\phi / C$, where $e^s_\phi$ is defined in Eq. 8 and $C$ is the number of latent channels. The reference error uses the same normalization. We apply the DPO objective to these channel-averaged errors with a preference scale of $\beta_{\text{impl}} = 5000$. Under the channel-summed error definition in the main text, this corresponds to $\beta = \beta_{\text{impl}}/C$.

### C.2 Trainable Parameters and Reference Model

**Adapter composition.** The SFT reasoner consists of the pretrained Qwen-Image-Edit-2511 model and a rank-64 LoRA adapter. During DPO, we freeze both components and introduce an additional rank-16 LoRA adapter. The policy activates both adapters, whereas the reference activates only the SFT adapter. Only the additional DPO adapter is optimized. Its initial contribution is zero, so the policy initially reproduces the SFT reference. The adapters remain separate from the pretrained weights during DPO training and inference.

For downstream correction, the SFT and SFT+DPO reasoners use the same SFT-adapted editor to execute their visual thoughts.

## D Experimental Prompt Templates

We present the task prompts used in our experiments. The placeholder {`description`} denotes the target image description; {`width`} and {`height`} denote the image dimensions after preprocessing.

### D.1 Visual Reasoning for Mismatch Localization

**Image output.** Our Visual Reasoner and the image-output baselines use the following prompt to generate a mask-annotated image.

**Mismatch Localization: Image Output**

```
Find the regions in the image that do not match the "{description},"
    segment them, and highlight them in solid black.
```

**Text output.** Qwen3-VL, BAGEL, and SenseNova-U1 use the following grounding instruction and user prompt to output bounding-box coordinates.

**Mismatch Localization: Text Output**

```
System instruction:
You are a precise visual grounding model. Given an image and a
    target text description, identify image regions that contradict
    the description. Do not include explanations in the final answer;
    return only the requested JSON object.

User prompt:
Find the region in the image that does not match the target
    description: "{description}". Return only one JSON object with
    this schema: {"box": [x1, y1, x2, y2]}. The box must tightly
    enclose the mismatched region, using normalized coordinates in
    [0, 1] where (0,0) is the top-left and (1,1) is the bottom-right.
    If there is no mismatch, return {"box": []}. Do not segment the
    region.

Image size after preprocessing: width={width}, height={height}. Use
    normalized coordinates, not pixel coordinates.
```

### D.2 Visual Reasoning for Mismatch Correction

**Implicit methods and closed-source models.** The implicit baselines, GPT-Image-2, and Nano Banana Pro receive the erroneous image with the same direct-correction prompt.

**Mismatch Correction: Direct Editing**

```
Identify and edit the regions in the image that are inconsistent
    with the following description: {description}. Modify them to
    match the description accurately.
```

**Explicit Text.** BAGEL, SenseNova-U1, and MLLM+Edit Model first use the text-output template in Section D.1. The predicted boxes are rendered as black regions on the input image, which is then passed to the editor with the correction template in Section D.3. ReasonEdit uses the direct-correction prompt above with its native thinking and reflection modes enabled.

**Explicit Image.** Our method and the image-based variants of BAGEL and SenseNova-U1 first use the image-output template in Section D.1. The generated mask-annotated image is then passed to the editor with the correction template in Section D.3.

### D.3 Correction with Provided Masks

All methods receive a ground-truth mask-annotated image and use the following correction prompt. The same template is used in the editing stage of the Explicit Text and Explicit Image pipelines above.

**Correction with Provided Masks**

```
The pure black areas (0,0,0) mark regions that are inconsistent with
    the target description. Regenerate these black regions to align
    with the description below, and preserve the rest of the image.
    Description: {description}
```

## E WHY DPO AFTER SFT: FIDELITY OF VISUAL THOUGHTS

SFT teaches the model to identify inconsistencies between an image and its text prompt and mark the corresponding regions with pure-black overlays. However, generating a visually plausible black overlay does not always yield a faithful binary mask. Figure 9 illustrates three residual artifacts after SFT and their refinement through DPO. We compare SFT and SFT+DPO on the same benchmark inputs.

**Consistent mask extraction.** Let $x$ denote the input image and $z$ a ground-truth or generated mask-overlay image. Following the evaluation implementation, we extract the mask at pixel $p$ as

$$\mathcal{M}(z; x)(p) = \mathbf{1}[z(p) = (0, 0, 0)]\, \mathbf{1}[x(p) \neq (0, 0, 0)]\,. \tag{12}$$

This rule selects exact-black pixels while excluding pixels that are already exact black in the input. It is applied identically to the ground truth, SFT output, and SFT+DPO output. For display, selected pixels are black and the background is white. No threshold relaxation, hole filling, or outlier removal is applied.

**Near-black regions.** In the top row, parts of the SFT overlay appear black but contain nonzero RGB values. These pixels are excluded by exact-black extraction, leaving holes within the intended mask. DPO reduces this internal fragmentation, producing a more complete binary region.

**Mask boundaries.** In the middle row, the binary mask extracted from the SFT output exhibits a fragmented boundary. DPO reduces these gaps and produces a more continuous contour, as shown in the enlarged views.

**Isolated black outliers.** In the bottom row, the SFT output introduces small exact-black components outside the target region. The enlarged background crop contains no exact-black pixels in the input or ground truth, so the highlighted component is an output artifact. This component is absent after DPO, illustrating a reduction in spurious mask activations.

Together, these examples demonstrate the value of DPO in refining the visual thought representation learned through SFT. By reducing near-black artifacts, improving boundary continuity, and suppressing isolated black outliers, DPO improves adherence to the mask-annotated output protocol and produces more faithful binary masks.

## F BEHAVIOR ON ERROR-FREE IMAGES

We evaluate our reasoner on the 100 original, error-free images corresponding to our benchmark, using the same target descriptions. For each output, we measure the proportion of newly generated pure-black pixels relative to the total image area, excluding pixels already pure black in the input.

The newly marked regions occupy only 1.57% of the image area on average. In 79% of the examples, this proportion is at most 1%, and in 97% of the examples, it is at most 10%. These results show that our reasoner typically produces empty or small masks on already-correct inputs, indicating that it generally avoids unnecessary error localization when the image satisfies the target description.

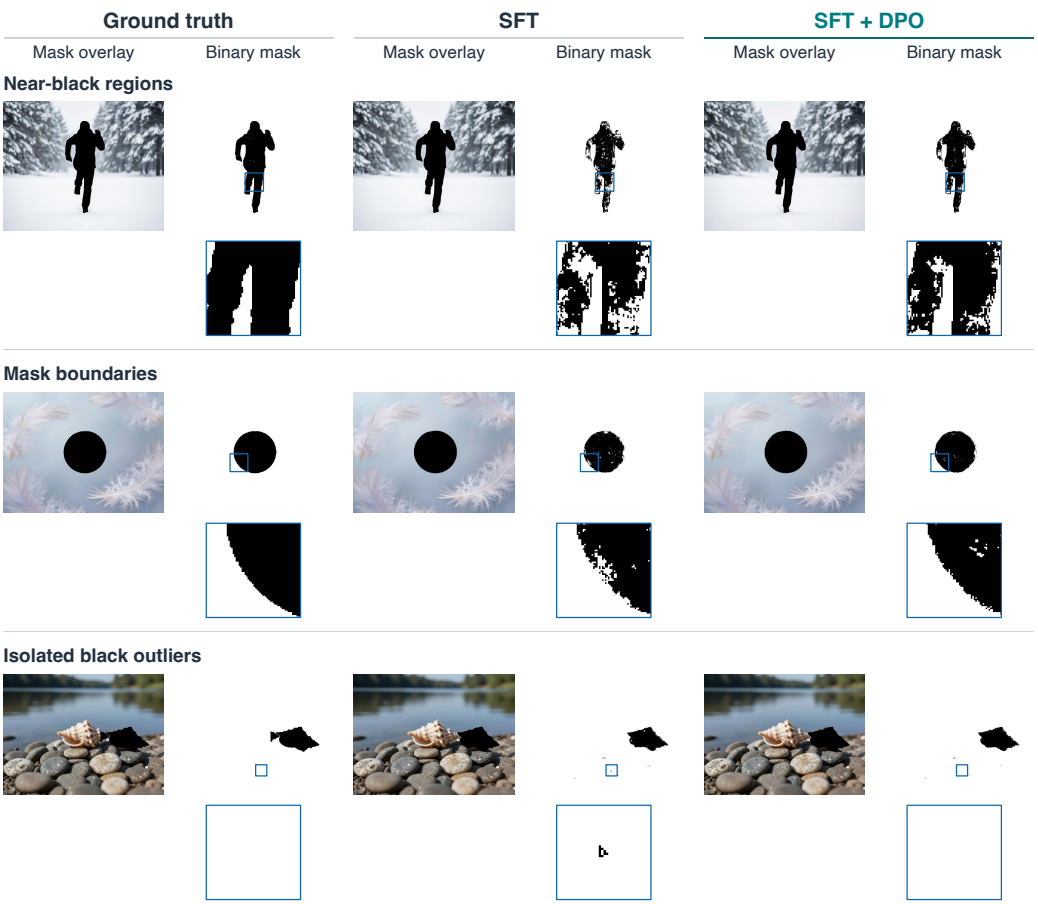



Figure 9: **Qualitative effects of DPO on visual thought fidelity.** From top to bottom: near-black regions, fragmented boundaries, and isolated black outliers. Each pair of columns shows a mask-overlay image and its extracted binary mask for ground truth, SFT, and SFT+DPO, respectively. Blue boxes mark the regions shown in the enlarged views below.

# G Additional Dataset Details

## G.1 Error Types and Visual Examples

Our construction of `VRB-Dataset` uses 24 error types to describe inconsistencies between an image and its target description. These types cover object presence, count and identity, visual attributes, pose and anatomy, geometric and structural properties, spatial relationships, and lighting and physical effects. Tab. 5 summarizes their definitions. Fig. 10 and Fig. 11 illustrate all 24 types using paired reference images and images with introduced errors.

Figs. 12 and 13 present five examples from `VRR-Dataset`, each with eight candidate visual reasoning images, their extracted binary masks, and the VLM-selected winner and loser.

## G.2 Dataset Sizes and Error-Type Distributions

The construction pipeline produces 48,421 valid records, of which 47,421 are used for `VRB-Dataset`. From the remaining 1,000 held-out records, we curate a 100-example benchmark by considering error-type diversity and mask annotation quality, selecting representative cases for evaluating visual reasoning.

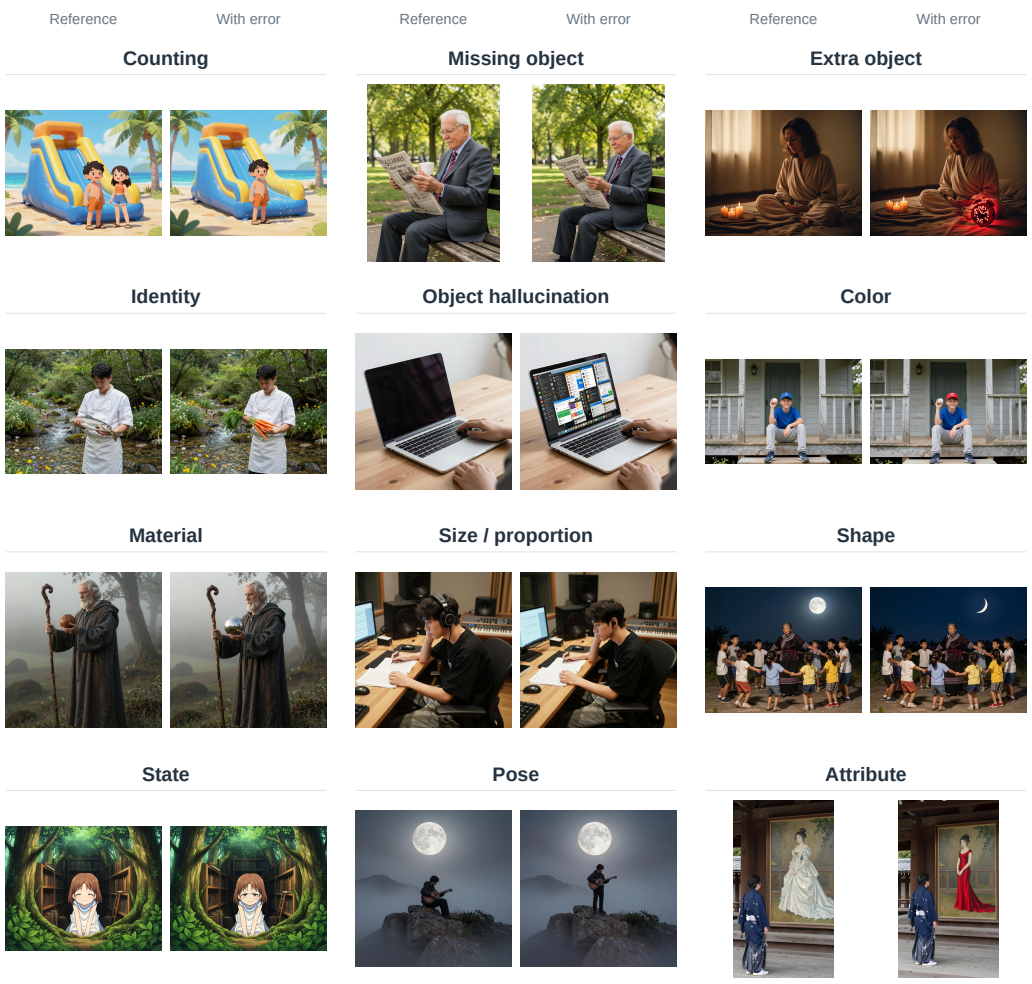



Figure 10: **Examples of constructed error types.** Each pair shows the reference image on the left and its counterpart with an introduced error on the right.

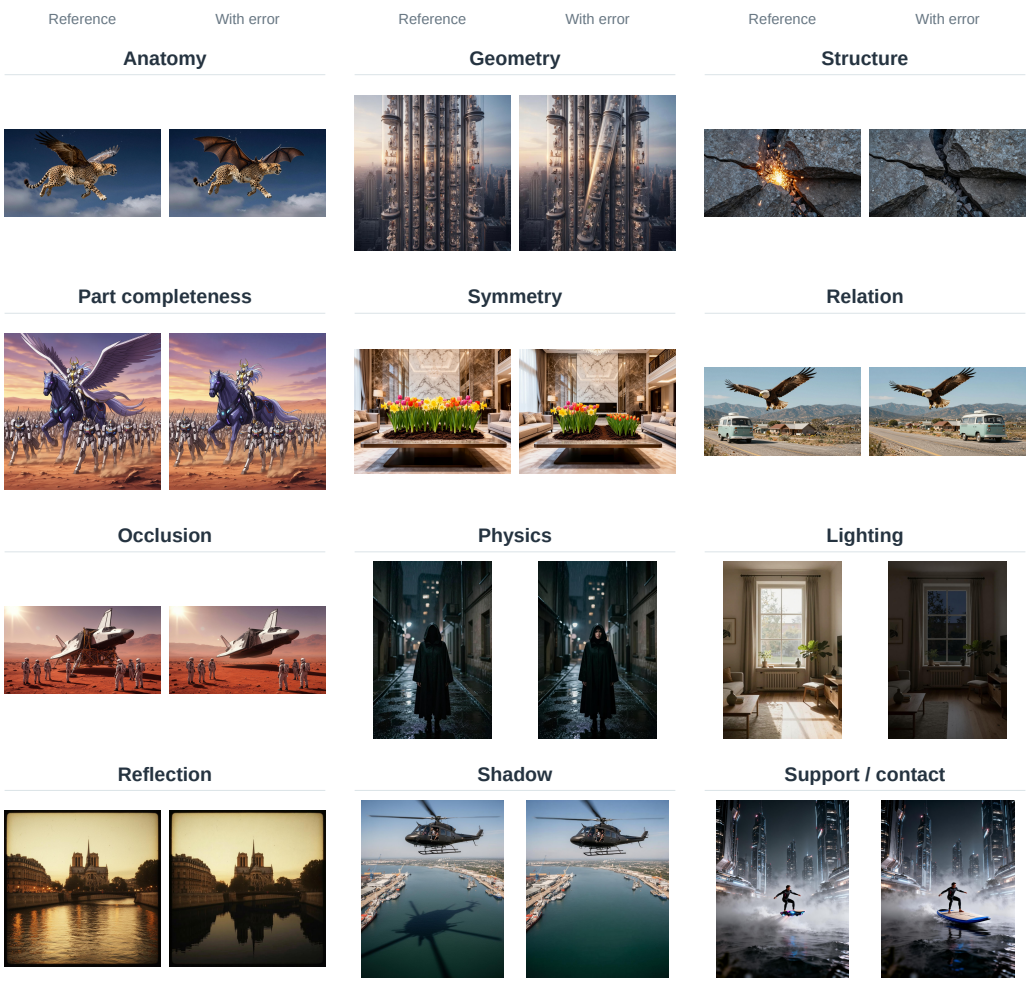
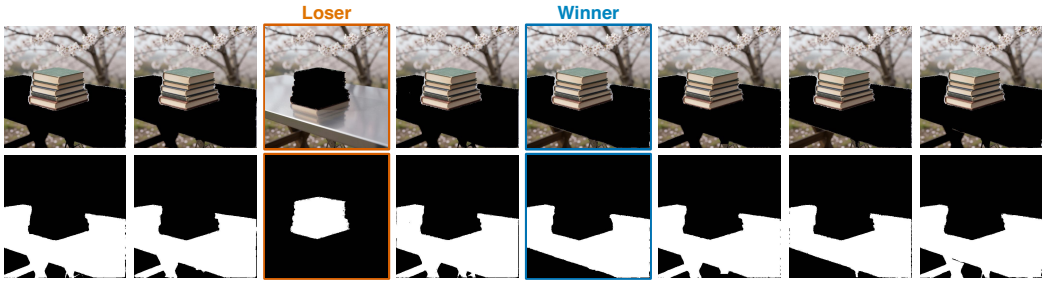
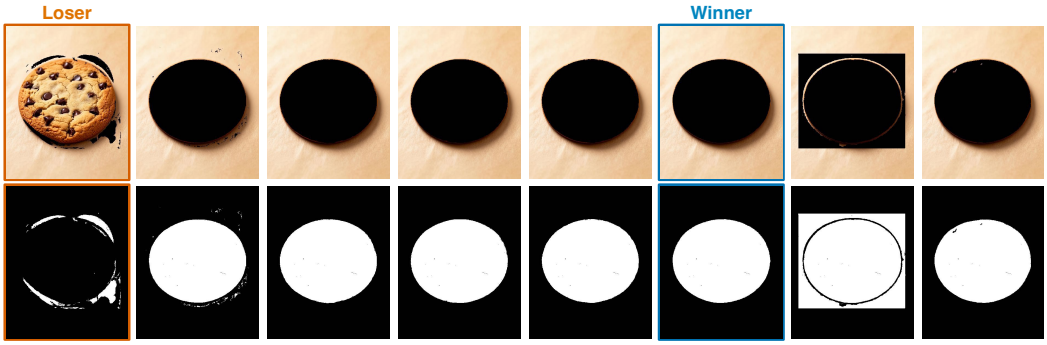


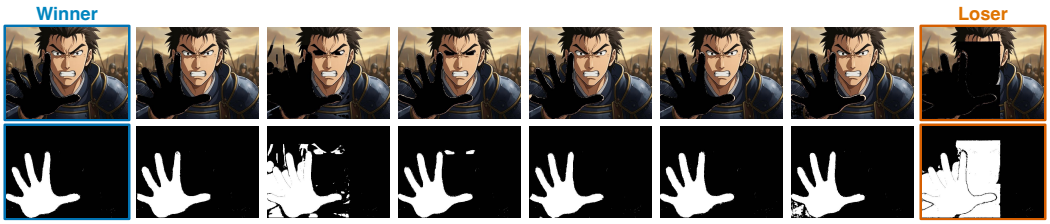

Figure 11: **Examples of the remaining error types.** Each pair shows the reference image on the left and its counterpart with an introduced error on the right.




*A stack of vintage books lies on a **wooden table** near a blooming cherry blossom tree.*




*I need an image of a freshly baked chocolate chip cookie **cut into a heart shape**.*




*A close-up of a serious anime warrior with his **fist clenched** and his face scrunched up in intense concentration before battle.*

Figure 12: **Examples from `VRR-Dataset`.** Each example shows eight candidate visual reasoning images (top), their extracted binary masks (middle), and the target description (bottom). White mask pixels correspond to pure-black candidate pixels after excluding those already pure black in the input image. Blue and orange outlines mark the VLM-selected winner and loser, respectively. The mismatched phrase is highlighted in bold red.

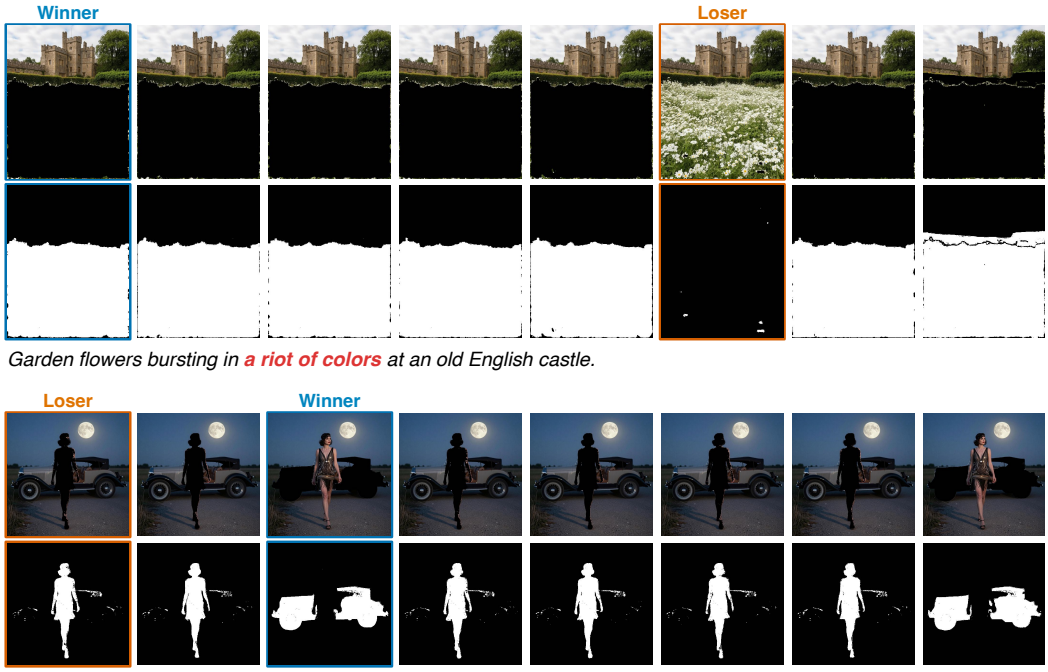



Figure 13: **Additional examples from `VRR-Dataset`.** Each example shows eight candidate visual reasoning images (top), their extracted binary masks (middle), and the target description (bottom). Blue and orange outlines mark the VLM-selected winner and loser, respectively. Mask extraction and text highlighting follow Fig. 12.

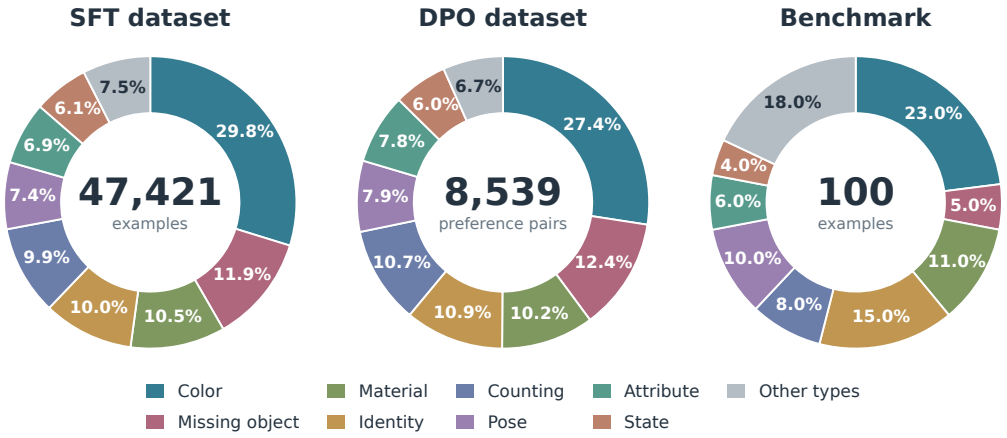



Figure 14: **Error-type distributions.** From left to right: `VRB-Dataset`, `VRR-Dataset`, and benchmark. Center labels indicate dataset sizes, and sector labels indicate percentages within each set. The same eight error types are shown individually across all panels; the remaining 16 types are grouped as *Other types*. DPO counts refer to preference pairs.

**`VRR-Dataset` selection.** We use the first 10,000 records of `VRB-Dataset` as the source pool for constructing `VRR-Dataset`, without explicitly balancing error types. We exclude no-change records and records without source mask overlays. For each eligible input, we generate eight candidate visual reasoning images and use a VLM to select a winner and a loser. We retain only examples with complete candidate sets and valid VLM selections. To filter out negligible masks and insufficiently distinct pairs, we discard a pair if the winner mask or the union of the two masks contains fewer than 64 pixels, or if their symmetric difference covers less than 5% of their union. These checks use pure-black mask pixels after excluding pixels already pure black in the input image. The resulting `VRR-Dataset` contains 8,539 preference pairs.

**Distribution statistics.** Fig. 14 presents the error-type distributions of the `VRB-Dataset`, `VRR-Dataset`, and benchmark.

## H Qualitative Comparison with All Baselines

This section extends the qualitative comparisons in the main paper to include all evaluated baselines. Fig. 15 and Fig. 16 present mismatch localization results, while Fig. 17 and Fig. 18 present mismatch correction results. Throughout these figures, bold red text highlights the relevant description content, and green outlines highlight `Seenk`.

## I Qualitative Results across Error Types

Figs. 19, 20, and 21 illustrate our method's ability to address diverse error types. The SFT+DPO Reasoner generates a visual thought that guides the SFT Editor to correct the identified inconsistency.

| Error type | Definition | SFT | DPO | Benchmark |
|---|---|---|---|---|
| Counting | Incorrect number of object instances. | 4,689 | 916 | 8 |
| Missing object | A required object is absent. | 5,643 | 1,060 | 5 |
| Extra object | An unrequested object is present. | 294 | 46 | 5 |
| Identity | An object has the wrong identity or category. | 4,740 | 933 | 15 |
| Object hallucination | A specified object is replaced by another entity. | 28 | 6 | 2 |
| Color | Incorrect object color. | 14,120 | 2,343 | 23 |
| Material | Incorrect material or surface appearance. | 4,974 | 875 | 11 |
| Size / proportion | Incorrect object size or proportions. | 633 | 120 | 3 |
| Shape | Incorrect object shape. | 742 | 129 | 1 |
| State | Incorrect object state, such as on or off. | 2,894 | 509 | 4 |
| Pose | Incorrect pose or action of a person or animal. | 3,508 | 671 | 10 |
| Attribute | Incorrect specified text, style, or other attribute. | 3,289 | 663 | 6 |
| Anatomy | Incorrect anatomy of a person or animal. | 51 | 6 | 1 |
| Geometry | Incorrect geometric properties or dimensionality. | 39 | 6 | 0 |
| Structure | Incorrect structural connections or organization. | 67 | 12 | 0 |
| Part completeness | A required part or detail is missing or altered. | 260 | 54 | 0 |
| Symmetry | Incorrect object symmetry. | 18 | 2 | 1 |
| Relation | Incorrect spatial arrangement between objects. | 268 | 42 | 2 |
| Support / contact | Incorrect support or contact between objects. | 215 | 46 | 0 |
| Occlusion | Incorrect occlusion or visibility relationships. | 17 | 2 | 0 |
| Lighting | Incorrect illumination or lighting effects. | 618 | 57 | 1 |
| Reflection | Incorrect reflected content or appearance. | 244 | 34 | 2 |
| Shadow | Incorrect shadow pattern or effect. | 60 | 4 | 0 |
| Physics | A physical effect contradicts the intended scene. | 10 | 3 | 0 |
| **Total** | | **47,421** | **8,539** | **100** |

Table 5: **Error-type definitions and complete counts.** Errors are defined relative to the intended image content specified by the text prompt. SFT and benchmark counts denote examples; DPO counts denote preference pairs.

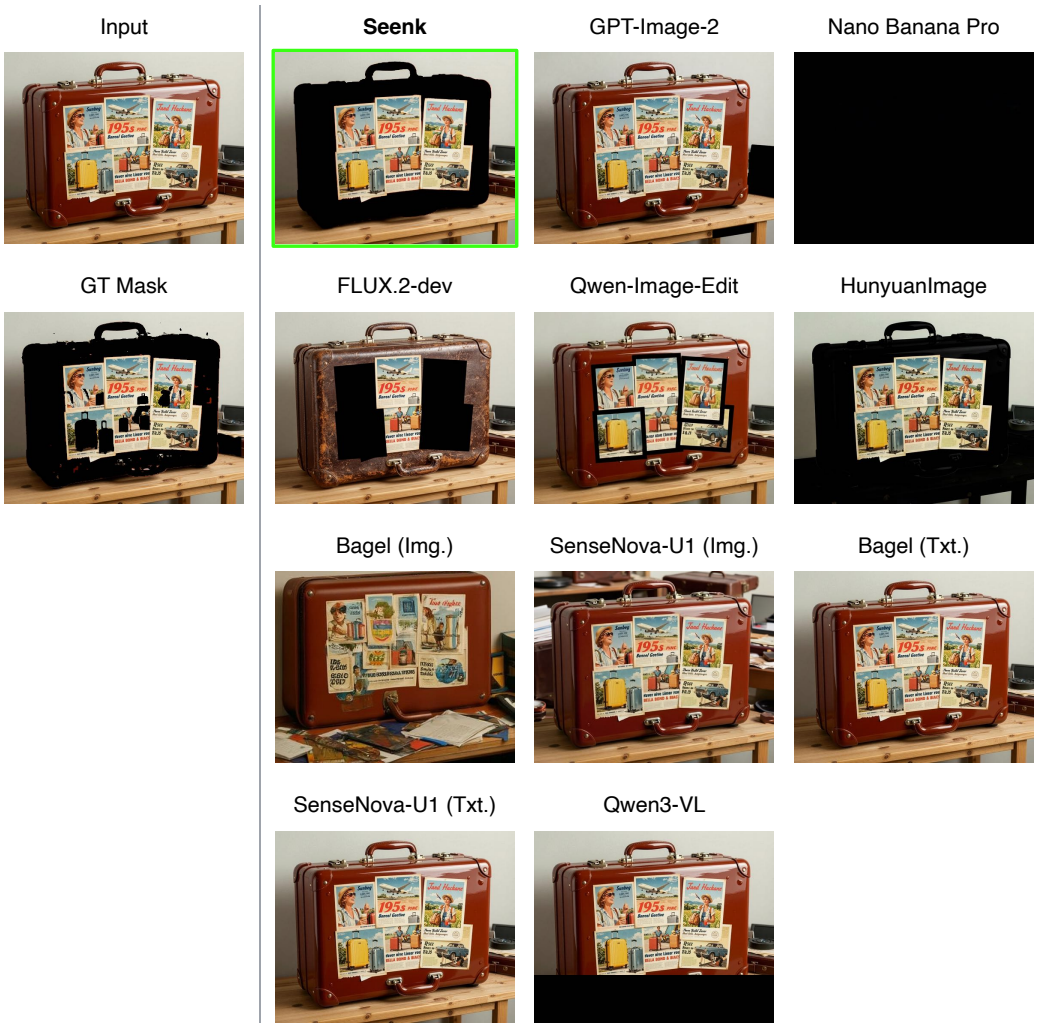



Figure 15: **Localizing a suitcase material mismatch.** The description specifies a leather-bound suitcase, whereas the input depicts a plastic suitcase. Predictions from `Seenk` and all baselines are shown alongside the input image and ground-truth mask.

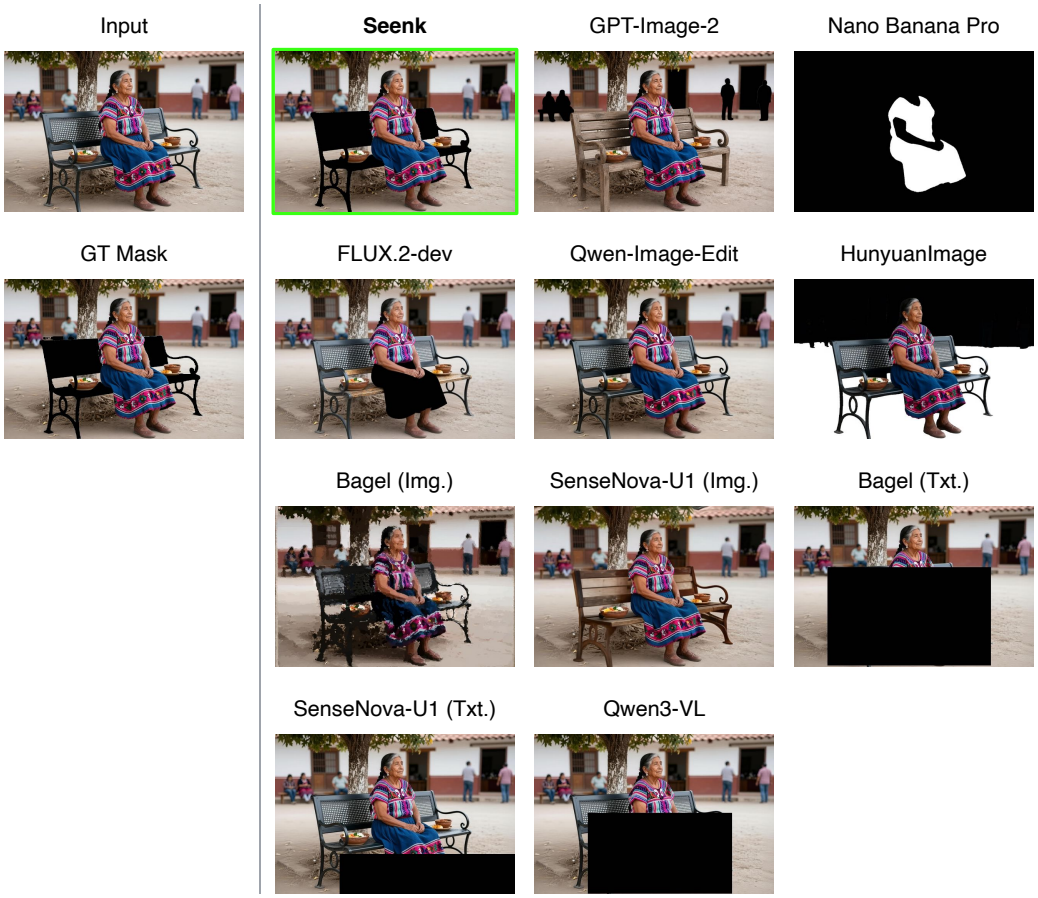



Figure 16: **Localizing a bench material mismatch.** The description specifies a wooden bench, whereas the input depicts a metal bench. Predictions from `Seenk` and all baselines are shown alongside the input image and ground-truth mask.




Figure 17: **Correcting a posture mismatch.** The input contains a standing child, whereas the description specifies that the children are sitting. Correction results from `Seenk` and all baselines are shown alongside the input and ground-truth image.

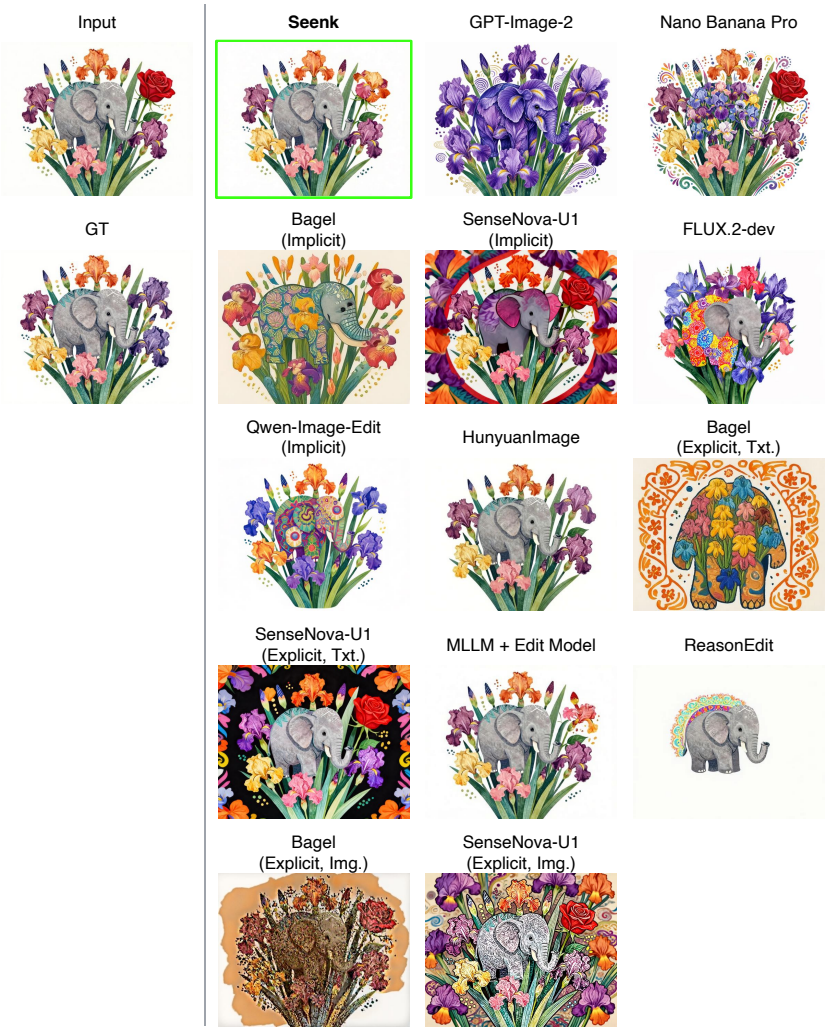



Figure 18: **Correcting a mismatch in flower type.** The input contains a rose among the irises, conflicting with the description of a bouquet of Japanese Irises. Correction results from `Seenk` and all baselines are shown alongside the input and ground-truth image.

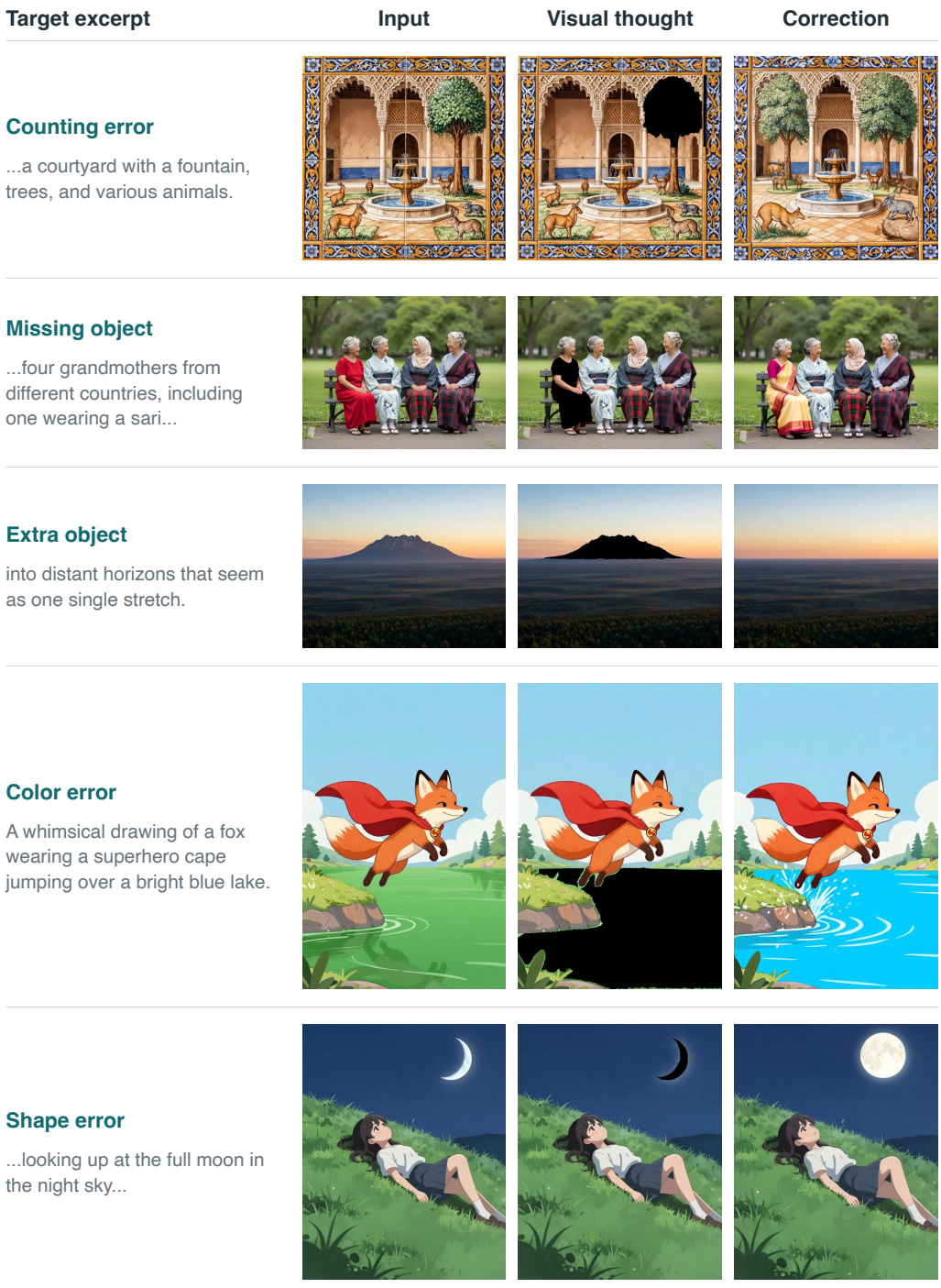



Figure 19: Qualitative results across error types. Each row shows an input image, the generated visual thought, and the corrected image, together with an excerpt from the target description.

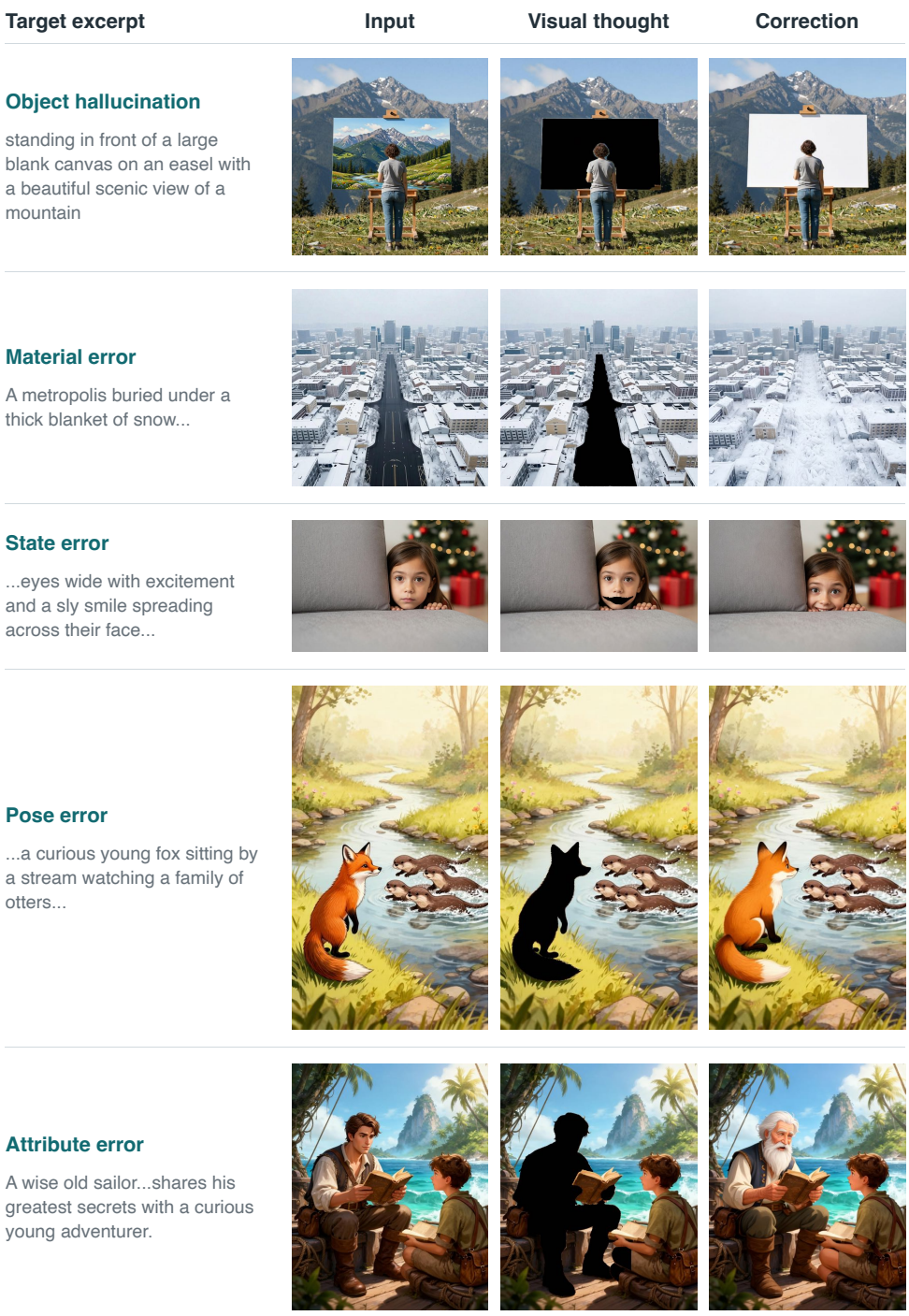



Figure 20: Additional qualitative results across error types, following the same layout as Fig. 19.

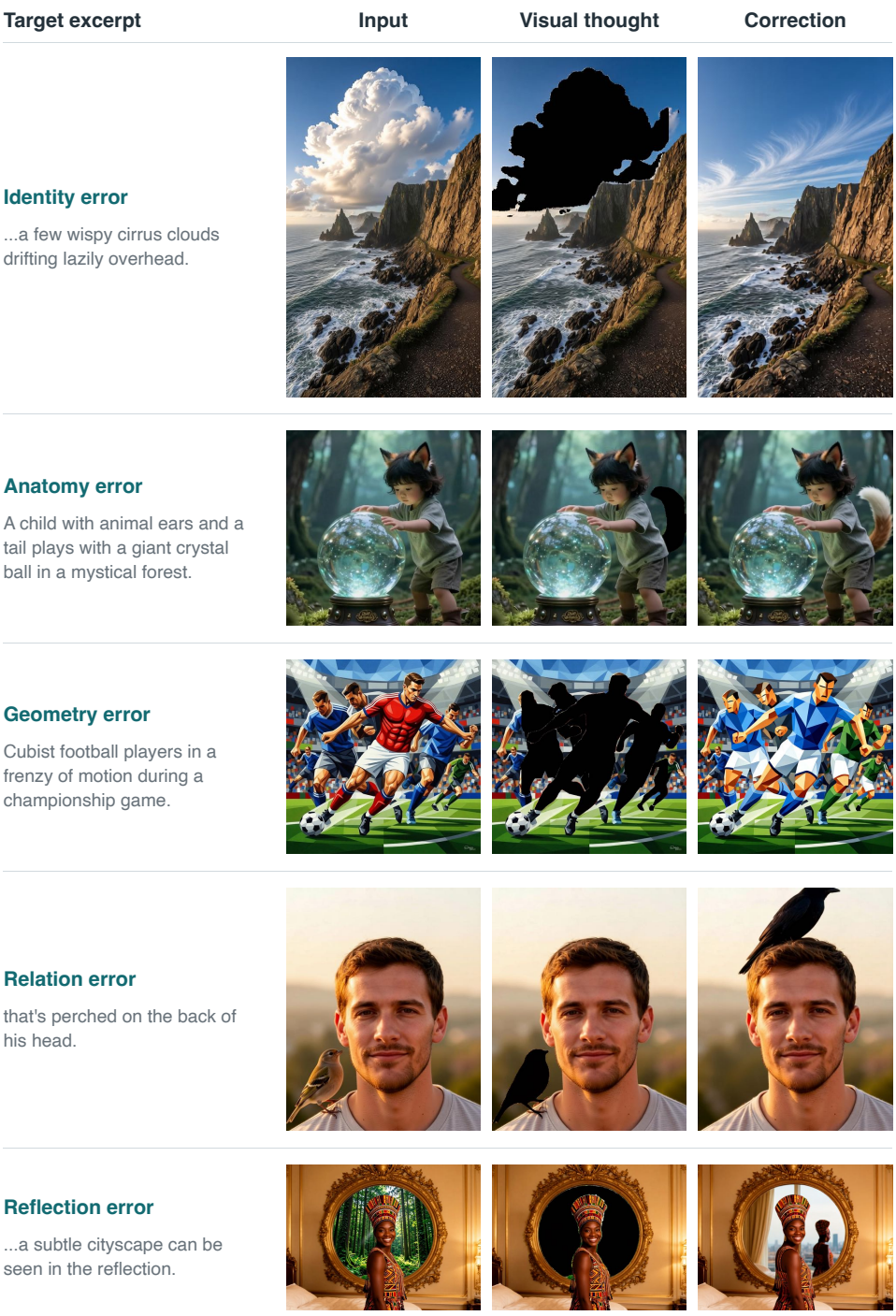



Figure 21: Further qualitative results across error types, following the same layout as Fig. 19.