# DREAMREALM: EXPLORING AND MATERIALIZING HALLUCINATION WORLDS FOR TRUSTWORTHY VISION-LANGUAGE MODELS

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

## ABSTRACT

Multimodal large language models (MLLMs) remain prone to visual hallucinations when correct responses depend on subtle visual details. One line of post-training methods constructs visual contrasts through image retrieval, generation, or editing to improve fine-grained perception. However, these contrasts are not built around the model's own hallucinations, making it difficult to ensure that supervision targets its actual perceptual confusions. Another line suppresses hallucinated outputs through response correction or preference learning under fixed image conditions, but leaves the design of supervision signals on the image-input side underexplored. Inspired by how humans correct misconceptions, we introduce **DreamRealm**, a post-training framework that explores diverse latent hallucinated responses, minimally edits original images to make them valid, and trains the model to assign each response to its supporting image. Our data construction pipeline, **RealmForgePipe**, mines observed and latent hallucinations from a target model and materializes them through image editing. Using hallucinations mined from Qwen3.5-9B, we construct the Realized Hallucination Dataset (**RealmSet**). Our training objective, **RealmMatch**, constructs a likelihood score matrix across image–response combinations and combines supervised learning with bidirectional contrastive losses to reinforce correct matches and suppress mismatches. Compared with the original Qwen3.5-9B model, **DreamRealm** improves performance across six visual perception and hallucination benchmarks, including gains of **7.76** percentage points in HallusionBench accuracy and **11.34** percentage points in MMVP pair accuracy. Together with ablation studies on data construction and training methods, these results demonstrate that hallucination assignment is a promising paradigm for improving visual perception and mitigating hallucinations. Our code is available anonymously at https://anonymous.4open.science/r/dreamrealm-3145.

## 1 INTRODUCTION

Recent multimodal large language models (MLLMs) have demonstrated strong capabilities in visual understanding (Bai et al., 2025; Wang et al., 2025). However, these models still generate content that contradicts image facts or lacks support from visual evidence, a phenomenon known as visual hallucination (Li et al., 2023; Guan et al., 2024; Wang et al., 2023). This problem is particularly pronounced when correct responses depend on subtle visual differences (Tong et al., 2024b; Guan et al., 2024). Such discrepancies between model outputs and visual evidence remain a major obstacle to the reliable application of multimodal models.

A line of post-training work mitigates visual hallucinations through response correction, contrastive supervision, and preference learning (Yu et al., 2024; 2025; Sarkar et al., 2025). These methods treat the image as a fixed condition and teach models to reject or revise hallucinated responses. Another line of work constructs visual contrasts through image generation or retrieval, helping models understand the relationship between fine-grained visual changes and different responses (Liu et al., 2026; Deng et al., 2026; Bai et al., 2026). However, these visual contrasts are not constructed specifically around the target model's hallucinations, making it difficult to ensure that supervision focuses on its actual perceptual confusions.



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

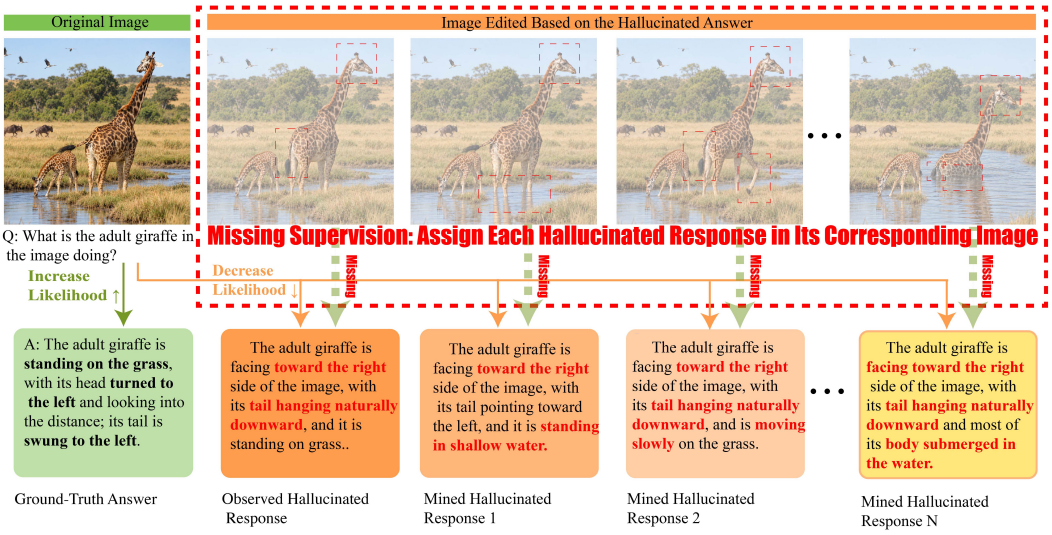



Figure 1: Motivation of DreamRealm. Existing post-training methods correct hallucinated responses, while supervision on the visual conditions that would make them valid remains underexplored. Inspired by how humans correct misconceptions, DreamRealm materializes these conditions and teaches the model to match hallucinated responses to their supporting images.

How humans correct misconceptions offers an intuitive starting point for designing supervision around specific perceptual confusions. When a child mistakes medicine tablets for foods such as candy or cookies, a parent may explain, "This is medicine, not a snack," and also teach the child what real candy, cookies, and other foods are. However, this form of supervision remains underexplored in existing post-training approaches, as illustrated in Figure 1. Inspired by this process, we explore a model's diverse latent hallucinated responses, construct visual conditions that support each response, and train the model to match these responses to their supporting visual conditions.

To this end, we introduce *DreamRealm*, a post-training framework centered on *matching hallucinated responses to their supporting images*. We first expose different error patterns through repeated high-temperature sampling from the target model, then expand the set of latent hallucinated responses by exploiting the enumerable key attributes of multimodal perception questions. We select hallucinated responses that the model is more inclined to produce based on their likelihood scores. An image-editing pipeline then minimally edits the original images to make the selected responses valid, materializing the model's hallucination world. We cross-pair the original and materialized images with the candidate responses to construct a likelihood score matrix. Through bidirectional contrastive losses, the model learns to suppress mismatched responses while assigning each hallucinated response to its corresponding supporting image. DreamRealm improves performance across six visual perception and hallucination benchmarks on Qwen3.5-9B (Qwen Team, 2026), including gains of 7.76 percentage points in HallusionBench accuracy and 11.34 percentage points in MMVP pair accuracy. We further conduct ablation studies on data construction and training methods to systematically analyze the roles of hallucination-guided image editing, latent hallucination mining, and bidirectional correspondence learning.

Our main contributions are as follows:

- **RealmForgePipe: A data pipeline for exploring and materializing a model's own hallucinations.** It derives image-editing targets from observed and mined latent hallucinations and constructs visual conditions that make these responses valid. Using this pipeline, we build the *Realized Hallucination Dataset* (**RealmSet**) from hallucinations mined from Qwen3.5-9B.
- **RealmMatch: learning to assign hallucinated responses to their supporting images.** We propose RealmMatch, which constructs a likelihood score matrix by cross-pairing original and materialized images with candidate responses. Through bidirectional contrastive

losses, it suppresses mismatched responses while learning to assign each hallucinated response to its corresponding supporting image.

- **Empirical validation and ablation analysis.** Our experiments and ablation studies demonstrate that hallucination assignment is a promising paradigm for improving visual perception and mitigating hallucinations. DreamRealm improves performance across six visual perception and hallucination benchmarks, with gains of 7.76 percentage points in HallusionBench accuracy and 11.34 percentage points in MMVP pair accuracy on Qwen3.5-9B. We further conduct ablations on data construction and training methods, providing detailed analyses of the contributions of individual components.

## 2 Related Work

### 2.1 Hallucination-aware Fine-tuning

Hallucination-aware fine-tuning improves the visual faithfulness of multimodal large language models through instruction augmentation, response correction, and preference learning. HALVA (Sarkar et al., 2025) constructs faithful and hallucinated response pairs and applies contrastive supervision to differing phrases to suppress visually unsupported content. RLAIF-V (Yu et al., 2025) builds preference data from model feedback and iteratively updates the model. Beyond direct response correction, REVERSE (Wu et al., 2026) introduces uncertainty tokens and retrospective resampling to identify and revise potentially hallucinated content. FINER-Tuning (Xiao et al., 2026) (FINER in our tables) introduces fine-grained contradictions concerning objects, attributes, and relations into otherwise correct descriptions to construct challenging negative queries, and employs DPO to mitigate the resulting hallucinations. These methods construct and leverage error signals in different ways, using contrastive supervision, preference optimization, or self-correction to distinguish visually grounded content from unsupported content. However, these methods primarily correct responses under a fixed image condition, generally giving insufficient consideration to the design of supervision signals on the image-input side.

### 2.2 Visual Contrast and Counterfactual Learning

Visual contrasts can be constructed by retrieving or pairing existing real images. These approaches use differences between images to help models distinguish the visual evidence supporting different responses. SymMPO (Liu et al., 2026) retrieves similar images using CLIP features and constructs corresponding responses for preference learning across visual conditions. CoCI (Manevich and Tsarfaty, 2025) uses similar images as contrastive inputs at inference time to adjust response distributions. Such approaches obtain visual contrasts from existing image collections, with the available contrasts depending on the images that can be retrieved or paired.

Another line of work uses image generation and editing to construct visual contrasts with controlled changes. IC-VCO (Deng et al., 2026) uses data from SymMPO to construct contrastive images through semantic editing for preference optimization. MED (Bai et al., 2026) generates minimal image-editing instructions conditioned on image content within predefined hallucination-prone semantic categories, and uses the resulting image pairs and fine-grained difference descriptions for supervised fine-tuning. These methods use generated visual differences to strengthen the connection between image content and model responses. Visual contrast is also used at inference time: visual contrastive decoding (VCD) (Leng et al., 2024) contrasts response distributions under different visual inputs to mitigate hallucinations. However, although these methods provide supervision on the image-input side, their visual contrasts are not specifically designed around the target model's hallucinations, making it difficult to ensure that supervision targets its actual perceptual confusions.

DreamRealm differs from these methods in two key aspects: (1) We explore a model's diverse latent hallucinated responses and learn to assign them to their supporting visual conditions, going beyond simply correcting erroneous responses under the original image. (2) We derive image-editing targets from the target model's own hallucinations, ensuring that the resulting visual contrasts directly address its perceptual confusions rather than relying on editing or generation that is not tailored to those confusions.

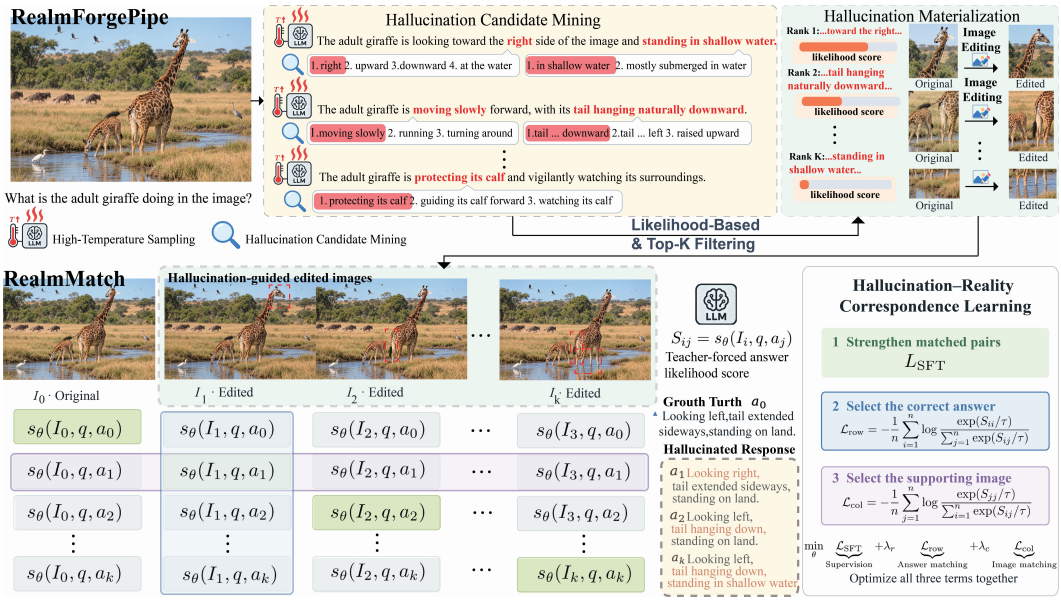



Figure 2: Overview of DreamRealm. **Top:** RealmForgePipe mines hallucinated responses through high-temperature sampling and key-attribute enumeration, selects candidates through likelihood ranking and Top-K filtering, and edits images to make the selected hallucinated responses valid. **Bottom:** RealmMatch constructs an image–response score matrix from original images with their ground-truth answers and edited images with their corresponding hallucinated responses. Through SFT on matched image–response pairs and bidirectional contrastive learning, it assigns hallucinated responses to their supporting images, learning the correspondence between responses and visual conditions.

# 3 METHOD

DreamRealm consists of RealmForgePipe, a data pipeline for exploring and materializing a model's own hallucinations, and RealmMatch, which learns image–response correspondence. Figure 2 illustrates the overall framework. RealmForgePipe comprises two stages: Section 3.1 introduces hallucination exploration, which collects observed hallucinations, mines latent hallucinations, and selects candidate responses through likelihood ranking; Section 3.2 introduces hallucination materialization, which uses minimal image editing and verification to construct visual conditions that make the selected hallucinated responses valid. The pipeline produces the *Realized Hallucination Dataset* (**RealmSet**), whose composition is described in Section 3.3. Finally, Section 3.4 presents RealmMatch, which combines supervised learning with bidirectional contrastive learning to assign hallucinated responses to their supporting images while retaining error-correction supervision under the original images.

## 3.1 EXPLORING THE LATENT HALLUCINATION WORLD

**Problem formulation.** Let $\mathcal{D} = \{(I_i, q_i, y_i^\star)\}_{i=1}^N$ denote a collection of seed examples, where $I_i$ is an image, $q_i$ is a perception question, and $y_i^\star$ is the ground-truth answer. For a target MLLM $M_\theta$, our goal is to identify a set $\mathcal{E}_i$ of incorrect responses that receive high likelihood under the model given $(I_i, q_i)$. We call these responses *latent perceptual hallucinations*. Incorrect responses produced under standard decoding are termed *observed hallucinations*; latent mining explores additional errors through high-temperature sampling and key-attribute enumeration.

**Diversified candidate elicitation.** We first use high-temperature sampling to explore diverse patterns of the model's hallucinated responses across perceptual dimensions. Given $(I_i, q_i)$, we sample $R$ responses at temperature $\tau_s$ to form

$$\mathcal{S}_i = \left\{ \widetilde{y}_i^{(r)} \sim p_\theta(\cdot \mid I_i, q_i; \tau_s) \right\}_{r=1}^{R}.$$

We then exploit the enumerable nature of key perceptual attributes to expand the candidate space. By enumerating alternative values of the attributes queried by $q_i$, we construct an additional candidate set $\mathcal{H}(q_i)$, extending coverage beyond the hallucination patterns captured by sampling.

We perform semantic deduplication over all proposed candidates, retaining only one representative from each group of semantically equivalent hallucinated responses to form the candidate pool $\mathcal{C}_i$.

**Likelihood-based hallucination ranking.** We score each candidate $e \in \mathcal{C}_i$ using teacher forcing. All candidates use the same answer template. If the rendered candidate contains tokens $z^e_{1:T_e}$, we define its log-likelihood score as the average token log-likelihood over the answer tokens:

$$s_i(e) = s_\theta(I_i, q_i, e) = \frac{1}{T_e} \sum_{t=1}^{T_e} \log p_\theta \left( z^e_t \mid I_i, q_i, z^e_{<t} \right).$$

Here $s_\theta(I, q, a)$ denotes the same mean token log-likelihood for any image $I$, question $q$, and response $a$; thus $s_i(e) = s_\theta(I_i, q_i, e)$. We rank the verified errors by $s_i(e)$ and retain the top $K_{\text{mine}}$ candidates, or all remaining candidates when fewer than $K_{\text{mine}}$ survive filtering:

$$\mathcal{E}_i = \text{Top}_{K_{\text{mine}};\, e \in \mathcal{C}_i} \; s_i(e).$$

We apply softmax normalization to the candidate scores to obtain the candidate-relative weights $\rho_i(e)$ within the candidate set:

$$\rho_i(e) = \frac{\exp(s_i(e))}{\sum_{e' \in \mathcal{C}_i} \exp(s_i(e'))}.$$

Importantly, $\rho_i(e)$ is not a calibrated absolute hallucination probability and depends on the coverage and granularity of $\mathcal{C}_i$. Figures 4 and 5 in Appendix B illustrate mined responses and their candidate-relative weights across action understanding, spatial relations, counting, clock reading, color recognition, and chart analysis. The selected set $\mathcal{E}_i$ is passed to the visual materialization stage, which constructs a hallucination-guided edited image supporting each hallucinated response.

### 3.2 Materializing Hallucinations through Verified Image Editing

RealmForgePipe constructs a supporting image for each selected response $y_m \in \mathcal{E}_i$. Semantic deduplication before editing ensures that equivalent hallucinated responses do not enter the same family. We prioritize minimal editing: changing only the visual content needed to make the target response valid while preserving unrelated scene content and structure as much as possible. Given $(I, q, y^\star, y_m)$, a verifier MLLM $\mathcal{V}$ specifies the content to preserve and modify and proposes three candidate instructions ordered by increasing expected edit extent. An image generation model attempts these instructions in order, with a verifier-assigned minimal-editing score of at least $4/5$ required alongside edit-success and visual-plausibility checks. The resulting image $I^\star$ undergoes quality checks and question-dependent validation; in strict blind checking, $\mathcal{V}$ answers the unchanged question from $I^\star$ without seeing $y_m$, and its answer must be semantically equivalent to $y_m$. The core validity requirement is

$$\text{Valid}(I, q, y_m) = 0, \qquad \text{Valid}(I^\star, q, y_m) = 1, \tag{1}$$

where $\text{Valid}(I, q, y)$ indicates whether response $y$ correctly answers question $q$ given image $I$. We further perform a within-family blind check for each $\mathcal{G} = (q, \{(I_i, a_i)\}_{i=0}^K)$, where $(I_0, a_0)$ is the original pair and the remaining pairs are edited. Without seeing the candidate responses, $\mathcal{V}$ answers the fixed question for each image; its answers are then semantically matched against the family responses to check mutual exclusivity:

$$\text{Valid}(I_i, q, a_j) = \mathbf{1}[i = j], \qquad \forall\, i, j \in \{0, \ldots, K\}. \tag{2}$$

Only families passing this blind check are used for bidirectional contrastive learning, with matched pairs as positives and off-diagonal combinations as negatives. Human review of individual edited image–response pairs provides a separate quality check. The per-edit generation, validation, and retry procedure is provided in Algorithm 1 in Appendix A, with the human review protocol in Appendix D.2.

### 3.3 REALMSET: THE REALIZED HALLUCINATION DATASET

We construct the *Realized Hallucination Dataset* (**RealmSet**) from VC-STaR's VisCoR-55K (Pan et al., 2026) using RealmForgePipe to materialize hallucinations mined from Qwen3.5-9B. Each question-centered family contains the original image–answer pair and human-approved edited image–response pairs, with the question held fixed. RealmSet spans 22 perceptual question categories and contains 28,960 human-approved edited examples. Training and held-out evaluation splits are separated by original-image family. Detailed construction statistics, category distributions, and split sizes are provided in Appendix D.

### 3.4 REALMMATCH: OFFLINE IMAGE–RESPONSE CORRESPONDENCE LEARNING

RealmMatch combines supervised learning with bidirectional contrast over image–response likelihoods, learning the supporting image for each response while retaining error correction under the original image.

**Constructing the image–response likelihood matrix.** For each verified family $\mathcal{G}$ with $K \geq 1$ edits, let $n = K + 1$ and define $S_{ij} = s_\theta(I_i, q, a_j)$ for $i, j = 0, \ldots, K$, using the mean token log-likelihood in Section 3.1. Here $a_0 = y^\star$ and $a_i$ for $i > 0$ is the hallucinated response used to construct $I_i$. Equation 2 makes diagonal pairs positive and off-diagonal pairs negative. We backpropagate through all scores to update $\theta$.

**Bidirectional correspondence learning.** We first introduce an image-conditioned response contrast loss (row-wise), which favors the matching response over other responses within each family, including hallucinated responses under their corresponding edited images:

$$\mathcal{L}_{\text{row}}^{\text{edit}} = -\frac{1}{n}\sum_{i=0}^{K} \log \frac{\exp(S_{ii}/\tau)}{\sum_{j=0}^{K}\exp(S_{ij}/\tau)}. \tag{3}$$

Correspondingly, the response-conditioned image contrast loss (column-wise) holds each response fixed and favors its supporting image over other images within the same family, encouraging each hallucinated response to be assigned to the edited image that makes it valid:

$$\mathcal{L}_{\text{col}}^{\text{edit}} = -\frac{1}{n}\sum_{j=0}^{K} \log \frac{\exp(S_{jj}/\tau)}{\sum_{i=0}^{K}\exp(S_{ij}/\tau)}. \tag{4}$$

Because the response wording is identical within each column, this term directly contrasts the visual conditions for the same response. The column softmax is a candidate-image classification objective constructed from conditional response scores, not a native image-generation probability. Here $\tau > 0$ is the contrastive temperature, distinct from the generation temperature $\tau_s$ used during mining.

We additionally apply SFT to matched image–response pairs to increase their likelihood:

$$\mathcal{L}_{\text{SFT}}^{\text{edit}} = -\frac{1}{n}\sum_{i=0}^{K} S_{ii}. \tag{5}$$

The joint training objective is

$$\mathcal{L}_{\text{edit}} = \lambda_r \mathcal{L}_{\text{row}}^{\text{edit}} + \lambda_c \mathcal{L}_{\text{col}}^{\text{edit}} + \mathcal{L}_{\text{SFT}}^{\text{edit}}, \tag{6}$$

where $\lambda_r, \lambda_c \geq 0$. All three terms are optimized jointly.

**Original examples without edited images.** For examples without an accepted edited image, bidirectional contrast reduces to row-wise contrast. We construct a record $\mathcal{U} = (I_0, q, a_0, \{b_j\}_{j=1}^{M})$, where $b_j$ are offline-verified incorrect responses and responses equivalent to $a_0$ are excluded. Supervised learning on the ground-truth answer is retained. Defining $s_+ = s_\theta(I_0, q, a_0)$ and $s_j^- = s_\theta(I_0, q, b_j)$, the objective is:

$$\mathcal{L}_{\text{plain}} = -s_+ + \lambda_r \log\left(1 + \sum_{j=1}^{M}\exp\left(\frac{s_j^- - s_+}{\tau}\right)\right). \tag{7}$$

The second term is the row-wise cross-entropy over the ground-truth answer and its $M$ negative responses. There is no column loss: only one image is available, and the incorrect responses have no corresponding positive image. We retain $\lambda_r$ unchanged rather than transferring the absent column weight to it. When $M = 0$, the contrastive term vanishes and the objective reduces to SFT. When $M = 1$, it becomes a binary logistic preference term plus SFT, without a reference-model likelihood ratio.

**Overall training objective.** The overall training objective takes two forms depending on whether accepted edited data are available. With accepted edits, bidirectional contrast assigns hallucinated responses to images that make them valid; otherwise, row-wise contrast suppresses these responses under the original image. Both cases retain supervised learning on correct image–response pairs. Let $K_g$ denote the number of accepted edits associated with record $g$. We average the corresponding losses over the $B$ records or families in each batch:

$$\mathcal{L}_{\text{batch}} = \frac{1}{B}\sum_{g=1}^{B}\mathcal{L}_g, \qquad \mathcal{L}_g = \begin{cases}\mathcal{L}_{\text{edit},g}, & K_g \geq 1,\\ \mathcal{L}_{\text{plain},g}, & K_g = 0.\end{cases} \tag{8}$$

We first average the losses within each family and then across the batch.

# 4 Experiments

## 4.1 Experimental Setup

**Training setup.** We use RealmForgePipe to mine and materialize Qwen3.5-9B's hallucinations into RealmSet, and train Qwen3.5-9B with RealmMatch on its training split (Appendix D). To examine model specificity, we reuse the same RealmSet training split, including the edited images and corresponding hallucinated responses constructed from Qwen3.5-9B, to train Qwen2.5-VL-7B (Bai et al., 2025), Gemma-4-E4B-it (Gemma Team, 2026), and InternVL3.5-8B (Wang et al., 2025) with RealmMatch, without rerunning RealmForgePipe for these models. Full training details are provided in Appendix G.

**Evaluation setup.** We evaluate visual perception and hallucination on six benchmarks. Evaluation details and inference results are provided in Appendix H.1.

## 4.2 Main Comparison

**Baseline sources.** We compare DreamRealm with specialized hallucination-mitigation methods, including VCD (Leng et al., 2024), CoCI-CAP (Manevich and Tsarfaty, 2025), RLAIF-V (Yu et al., 2025), SymMPO (Liu et al., 2026), mDPO (Wang et al., 2024a), VC-STaR (Pan et al., 2026), MED (Bai et al., 2026) , IC-VCO (Deng et al., 2026), FINER (Xiao et al., 2026), IRIS (Li et al., 2026), and $C^2$-DPO (Ko et al., 2026). More details are provided in the appendix J.

Table 1 compares DreamRealm with specialized hallucination-mitigation methods and examines the effect of aligning the hallucination-mining model with the target model. All DreamRealm variants use the same RealmSet constructed from Qwen3.5-9B hallucinations. Each backbone is trained separately and compared with its corresponding raw checkpoint, without backbone-specific hallucination re-mining or image editing.

**Training on own hallucinations.** We compare DreamRealm with the corresponding base model to control for backbone differences. On Qwen3.5-9B, DreamRealm improves all seven metrics, mitigating hallucinations while enhancing fine-grained perception. The gains are largest on Qwen3.5-9B, reflecting DreamRealm's model-specific design: RealmForgePipe constructs supervision from the model's high-likelihood hallucinations, aligning the training data with its perceptual confusions. As shown in Table 1, Qwen3.5-9B + DreamRealm obtains the best numerical value in six of the seven reported metrics under our unified evaluation protocol. The only exception is ObjectHal, where its performance remains close to the best result from RLAIF-V. Qualitative comparisons are shown in Figure 3, with additional examples in Appendix I.

Table 1: Main comparison on hallucination benchmarks. Block I reports specialized hallucination-mitigation methods, while Blocks II and III report matched and mismatched hallucination-mining and target-model settings, respectively.

| Model / Method | HallusionBench aAcc ↑ | MMVP Item Acc ↑ | MMVP Pair Acc ↑ | AMBER Avg Acc ↑ | MMHal-Bench Hall. Rate ↓ | CV-Bench-2D Macro Acc ↑ | ObjectHal CHAIRs ↓ |
|---|---|---|---|---|---|---|---|
| **I. Specialized hallucination-mitigation methods** | | | | | | | |
| VCD (Leng et al., 2024) | 55.54 | 63.00 | 34.00 | 77.71 | 54.17 | 52.42 | 44.33 |
| CoCI-CAP (Manevich and Tsarfaty, 2025) | 61.47 | 75.67 | 54.67 | 84.05 | 36.46 | 67.46 | 20.67 |
| RLAIF-V (Yu et al., 2025) | 55.45 | 57.33 | 24.67 | 80.59 | 39.58 | 59.26 | **8.33** |
| SymMPO (Liu et al., 2026) | 51.02 | 61.67 | 29.33 | 79.75 | 56.25 | 61.37 | 15.67 |
| mDPO (Wang et al., 2024a) | 60.94 | 64.67 | 31.33 | 80.25 | 45.83 | 53.73 | 14.00 |
| VC-STaR (Pan et al., 2026) | 67.85 | 76.33 | 56.00 | 84.20 | 44.79 | 74.71 | 12.33 |
| MED (Bai et al., 2026) | 61.91 | 79.00 | 60.00 | 82.02 | 31.25 | 69.29 | 9.00 |
| IC-VCO (Deng et al., 2026) | 68.47 | 75.00 | 52.67 | 85.88 | 39.58 | 70.04 | 18.00 |
| FINER (Xiao et al., 2026) | 64.39 | 77.33 | 55.33 | 85.74 | 32.29 | 76.29 | 19.33 |
| IRIS (Li et al., 2026) | 48.54 | 63.33 | 29.33 | 77.59 | 52.08 | 60.94 | 40.33 |
| $C^2$-DPO (Ko et al., 2026) | 47.92 | 64.33 | 32.00 | 76.95 | 44.79 | 60.87 | 16.67 |
| **II. Training on own hallucinations** | | | | | | | |
| Qwen3.5-9B (Raw) (Qwen Team, 2026) | 70.95 | 80.00 | 63.33 | 88.34 | 47.92 | 82.16 | 30.33 |
| **+ DreamRealm** | **78.21**$_{+7.26}$ | **86.67**$_{+6.67}$ | **74.67**$_{+11.34}$ | **91.01**$_{+2.67}$ | **27.08**$_{+20.84}$ | **82.89**$_{+0.73}$ | 8.67$_{+21.66}$ |
| **III. Training on another model's hallucinations** | | | | | | | |
| Qwen2.5-VL-7B (Raw) (Bai et al., 2025) | 67.40 | 77.33 | 56.00 | 85.22 | 38.54 | 75.37 | 13.33 |
| **+ DreamRealm** (Qwen3.5-mined) | 71.74$_{+4.34}$ | 78.33$_{+1.00}$ | 59.33$_{+3.33}$ | 86.14$_{+0.92}$ | 30.21$_{+8.33}$ | 75.94$_{+0.57}$ | 11.67$_{+1.66}$ |
| Gemma-4-E4B-it (Raw) (Gemma Team, 2026) | 65.37 | 71.33 | 46.67 | 83.67 | 37.50 | 65.29 | 20.33 |
| **+ DreamRealm** (Qwen3.5-mined) | 67.14$_{+1.77}$ | 71.67$_{+0.34}$ | 47.33$_{+0.66}$ | 85.29$_{+1.62}$ | 35.42$_{+2.08}$ | 65.08$_{-0.21}$ | 18.67$_{+1.66}$ |
| InternVL3.5-8B (Raw) (Wang et al., 2025) | 70.50 | 76.67 | 57.33 | 88.30 | 29.17 | 79.03 | 13.33 |
| **+ DreamRealm** (Qwen3.5-mined) | 71.21$_{+0.71}$ | 78.67$_{+2.00}$ | 60.00$_{+2.67}$ | 88.80$_{+0.50}$ | 28.13$_{+1.04}$ | 78.84$_{-0.19}$ | 11.00$_{+2.33}$ |

**Training on another model's hallucinations.** We further assess how performance gains change when the training data are not tailored to the target model's own hallucinations. Qwen3.5-9B serves as both the hallucination-mining model and the target model, whereas Qwen2.5-VL-7B, Gemma-4-E4B-it, and InternVL3.5-8B are trained on the same RealmSet constructed from Qwen3.5-9B hallucinations, without model-specific re-mining or image editing. The matched Qwen3.5-9B setting yields the largest gains on all seven metrics. Qwen2.5-VL-7B also improves consistently, which may reflect partially shared perceptual error patterns within the Qwen family. In contrast, the gains on Gemma-4-E4B-it and InternVL3.5-8B are smaller, and both models exhibit a slight regression on CV-Bench-2D. These results suggest that hallucination spaces overlap across models without being fully interchangeable.

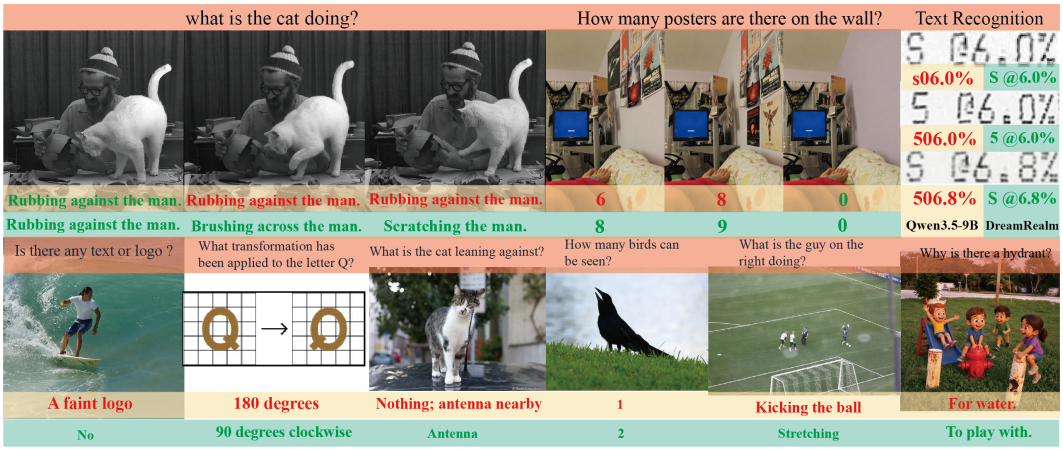



Figure 3: Selected examples from the held-out test set before and after DreamRealm training. Yellow/mint bands denote Qwen3.5-9B/DreamRealm; red/green text marks incorrect/correct answers. Responses are condensed.

### 4.3 Ablation Study

We separate the contributions of the RealmForgePipe data pipeline and the RealmMatch training method through two complementary ablations on Qwen3.5-9B: a data-construction ablation for RealmForgePipe and a training-method ablation for RealmMatch.

**RealmForgePipe data ablation.** Table 2 evaluates the components of RealmForgePipe with RealmMatch training and optimization fixed. The configurations are cumulative: Original Only uses unedited source examples; Unguided Edits adds randomly sampled edited examples; Observed Only constructs edits solely from incorrect responses produced by the mining model under standard decoding; Full RealmForgePipe additionally incorporates latent hallucinations discovered through high-temperature sampling and key-attribute enumeration. Original Only substantially degrades HallusionBench, while Unguided Edits improves several metrics but remains below Raw on HallusionBench and CV-Bench-2D. Observed Only outperforms Unguided Edits on five of seven metrics, and adding latent mining improves all seven metrics over Observed Only; Full RealmForgePipe consequently improves every metric over Raw. These results show that observed and latent hallucinations provide complementary supervision: observed errors align the edited data with the model's existing failure modes, while latent mining expands the coverage to plausible perceptual confusions that are not exposed by standard decoding.

Table 2: RealmForgePipe data ablation on Qwen3.5-9B. All non-Raw variants use the same RealmMatch training setup and differ only in their data and construction components.

| Configuration | HallusionBench aAcc ↑ | MMVP Item Acc ↑ | MMVP Pair Acc ↑ | AMBER Avg Acc ↑ | MMHal-Bench Hall. Rate ↓ | CV-Bench-2D Macro Acc ↑ | ObjectHal CHAIRs ↓ |
|---|---|---|---|---|---|---|---|
| Raw | 70.45 | 80.00 | 63.33 | 88.34 | 47.92 | 82.16 | 30.33 |
| Original Only | 61.82 | 80.33 | 62.67 | 87.21 | 35.42 | 79.63 | 11.00 |
| Unguided Edits | 69.35 | 83.67 | 69.33 | 90.97 | 34.38 | 80.37 | **8.00** |
| Observed Only | 73.87 | 86.00 | 74.00 | 90.04 | 30.21 | 81.86 | 13.33 |
| **Full RealmForgePipe** | **78.21** | **86.67** | **74.67** | **91.01** | **27.08** | **82.89** | 8.67 |

**RealmMatch: training-method ablation.** Table 3 compares training methods under the same model initialization and RealmForgePipe data. SFT improves all seven metrics and achieves the best AMBER result, confirming the value of the constructed data. DPO performs well on MMVP but degrades HallusionBench, while GRPO provides more balanced gains but remains behind RealmMatch. RealmMatch improves all seven metrics and achieves the best result on six. Unlike the compared baselines, RealmMatch applies its full bidirectional correspondence objective to structured image–response families. Its overall advantage shows that the complete RealmMatch formulation extracts more effective supervision from the same data than the compared training objectives.

Table 3: RealmMatch training-method ablation on Qwen3.5-9B. All trained variants use the same data produced by Full RealmForgePipe and differ only in their training objectives.

| Training method | HallusionBench aAcc ↑ | MMVP Item Acc ↑ | MMVP Pair Acc ↑ | AMBER Avg Acc ↑ | MMHal-Bench Hall. Rate ↓ | CV-Bench-2D Macro Acc ↑ | ObjectHal CHAIRs ↓ |
|---|---|---|---|---|---|---|---|
| Raw (no training) | 70.45 | 80.00 | 63.33 | 88.34 | 47.92 | 82.16 | 30.33 |
| SFT | 73.16 | 83.67 | 68.67 | **92.12** | 35.42 | 82.85 | 12.33 |
| DPO | 67.67 | 86.00 | 74.00 | 89.46 | 34.38 | 78.81 | 11.33 |
| GRPO | 74.31 | 84.33 | 70.00 | 90.65 | 30.21 | 82.16 | 15.67 |
| **RealmMatch (ours)** | **78.21** | **86.67** | **74.67** | 91.01 | **27.08** | **82.89** | **8.67** |

## 5 Conclusion

We introduce DreamRealm, a post-training framework that materializes model hallucination worlds and uses contrastive learning to assign hallucinated responses to their corresponding visual inputs. Consistent improvements across multiple benchmarks demonstrate the potential of this hallucination assignment paradigm to enhance visual perception and mitigate visual hallucinations.

## 6 AI USE STATEMENT

In this work, we used generative AI tools for synthetic-data construction and manuscript preparation. Within RealmForgePipe, Qwen3.5-9B generated candidate hallucinated responses, GPT-Image-2 produced hallucination-guided image edits, and Gemini 2.5 Pro assisted with sample screening, edit planning, and automated blind verification. All HG-Edited examples retained in the final dataset were subsequently reviewed by human annotators under the protocol described in Appendix D.2. We also used OpenAI ChatGPT to assist with English editing. The authors reviewed all AI-assisted text against the underlying methods, experimental outputs, and cited sources, and revised or rejected suggestions when necessary. Generative AI outputs were not treated as independent scientific evidence, and the authors made all final methodological and interpretive decisions. We take full responsibility for the final content of this work, including text, claims, data, and artifacts produced with the assistance of generative AI.

## REFERENCES


Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., ... & Lin, J. (2025). Qwen2.5-VL technical report. CoRR abs/2502.13923 (2025). arXiv preprint arXiv:2502.13923.

Bai, T., Fan, Y., Jiantao, Q., Sun, F., Song, J., Han, J., ... & Yuan, B. (2026). Hallucination at a glance: Controlled visual edits and fine-grained multimodal learning. Advances in Neural Information Processing Systems, 38, 135360-135393.

Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., ... & Zhao, F. (2024). Are we on the right way for evaluating large vision-language models?. Advances in Neural Information Processing Systems, 37, 27056-27087.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., ... & Velic, M. (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261.

Deng, H., Zou, X., Jin, Z., Chen, C., Lu, H., & Hu, X. (2026). Learning from Fine-Grained Visual Discrepancies: Mitigating Multimodal Hallucinations via In-Context Visual Contrastive Optimization. arXiv preprint arXiv:2605.31312.

Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., ... & Chen, K. (2024). VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In Proceedings of the 32nd ACM International Conference on Multimedia (pp. 11198–11201).

Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., ... & Zhou, T. (2024, June). Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 14375-14385). IEEE.

Ko, B., Park, J., Kim, J., Na, J., Cho, J., & Kim, H. J. (2026, September). Context Blindness in DPO: Mitigating Object Hallucination in MLLMs via Context-Calibrated Preference Optimization. In European Conference on Computer Vision (pp. 40-58). Cham: Springer Nature Switzerland.

Leng, S., Zhang, H., Chen, G., Li, X., Lu, S., Miao, C., & Bing, L. (2024, June). Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 13872-13882). IEEE.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, X., & Wen, J. R. (2023, December). Evaluating object hallucination in large vision-language models. In Proceedings of the 2023 conference on empirical methods in natural language processing (pp. 292-305).

Li, Y., Yan, Y., Han, J., Ming, F., Lv, L., & Jin, Y. (2026). IRIS: Implicit Reward-Guided Internal Sifting for Mitigating Multimodal Hallucination. arXiv preprint arXiv:2602.01769.

Liu, W., Song, X., Li, J., Wei, Y., Zheng, N., Yin, J., & Nie, L. (2026). Mitigating hallucination through theory-consistent symmetric multimodal preference optimization. Advances in Neural Information Processing Systems, 38, 111259-111284.

Manevich, A., & Tsarfaty, R. (2025, August). Making LVLMs Look Twice: Contrastive Decoding with Contrast Images. In Proceedings of the 1st Workshop on Multimodal Augmented Generation via Multimodal Retrieval (MAGMaR 2025) (pp. 65-78).

Neuhaus, Y., & Hein, M. (2025). Repope: Impact of annotation errors on the pope benchmark. arXiv preprint arXiv:2504.15707.

Pan, Z., Wu, Y., Hua, J., Feng, J., Yan, S., Deng, B., ... & Ye, J. (2026). Through the Lens of Contrast: Self-Improving Visual Reasoning in VLMs. arXiv preprint arXiv:2603.02556.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

Qiu, Z., Wang, Z., Li, X., Li, Y., Xu, Y., Wang, Y., ... & Liu, D. (2026). On the Design of Qwen3.8-Next Architecture: Evaluation, Efficiency, and Training Stability. arXiv preprint arXiv:2608.30320.

Qwen Team. (2026). *Qwen3.5: Towards native multimodal agents*. Qwen Blog. `https://qwen.ai/blog?id=qwen3.5`

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36, 53728-53741.

Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2020, November). Zero: Memory optimizations toward training trillion parameter models. In SC20: international conference for high performance computing, networking, storage and analysis (pp. 1-16). IEEE.

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., & Saenko, K. (2018). Object hallucination in image captioning. In Proceedings of the 2018 conference on empirical methods in natural language processing (pp. 4035-4045).

Sarkar, P., Ebrahimi, S., Etemad, A., Beirami, A., Arik, S., & Pfister, T. (2025, May). Mitigating object hallucination in mllms via data-augmented phrase-level alignment. In International Conference on Learning Representations (Vol. 2025, pp. 73796-73846).

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., ... & Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., ... & Darrell, T. (2024, August). Aligning large multimodal models with factually augmented rlhf. In Findings of the association for computational linguistics: ACL 2024 (pp. 13088-13110).

Team, G., Abd, S. E., Aggarwal, V., Algayres, R., Andreev, A., Bachem, O., ... & Geisler, S. (2026). Gemma 4 technical report. arXiv preprint arXiv:2607.02770.

Team, K., Bai, T., Bai, Y., Bao, Y., Cai, J., Cai, X., ... & Liu, Y. (2026). Kimi k3: Open frontier intelligence. arXiv preprint arXiv:2607.24653.

Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M., Akula, S. C., ... & Xie, S. (2024a). Cambrian-1: A fully open, vision-centric exploration of multimodal llms. Advances in Neural Information Processing Systems, 37, 87310-87356.

Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., & Xie, S. (2024b, June). Eyes wide shut? exploring the visual shortcomings of multimodal llms. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 9568-9578). IEEE.

Wang, F., Zhou, W., Huang, J. Y., Xu, N., Zhang, S., Poon, H., & Chen, M. (2024a, November). mdpo: Conditional preference optimization for multimodal large language models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (pp. 8078-8088).

Wang, J., Wang, Y., Xu, G., Zhang, J., Gu, Y., Jia, H., ... & Sang, J. (2023). Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. arXiv preprint arXiv:2311.07397.

Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., ... & Luo, G. (2025). Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265.

Wang, W., Ren, Y., Luo, H., Li, T., Yan, C., Chen, Z., ... & Dai, J. (2024b, September). The all-seeing project v2: Towards general relation comprehension of the open world. In European Conference on Computer Vision (pp. 471-490). Cham: Springer Nature Switzerland.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

Wu, T. H. P., Lee, H., Ge, J., Gonzalez, J., Darrell, T., & Chan, D. (2026). Generate, but verify: Reducing hallucination in vision-language models with retrospective resampling. Advances in Neural Information Processing Systems, 38, 65749-65777.

Xiao, R., Kim, S., Xian, Y., Akata, Z., & Alaniz, S. (2026). FINER: MLLMs Hallucinate under Fine-grained Negative Queries. arXiv preprint arXiv:2603.17662.

Yu, T., Yao, Y., Zhang, H., He, T., Han, Y., Cui, G., ... & Chua, T. S. (2024). Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13807-13816).

Yu, T., Zhang, H., Li, Q., Xu, Q., Yao, Y., Chen, D., ... & Sun, M. (2025, June). Rlaif-v: Open-source ai feedback leads to super gpt-4v trustworthiness. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 19985-19995). IEEE.

# A HALLUCINATION MATERIALIZATION ALGORITHM

Algorithm 1 provides the complete image-editing workflow described in Section 3.2, including difficulty tagging, quality verification, question-dependent blind checking, and bounded retry logic. Stages 1.5 and 2.5 record metadata only and never reject samples. Each instruction–type pair permits at most two synthesis attempts in total. After automated checks, human annotators review the candidate before curation; only human-approved records enter the final HG-Edited Data. Human-rejected candidates are excluded.

**Minimal-editing principle.** Minimal editing prioritizes changes necessary to make the target response valid while preserving unrelated scene content and structure. The planner specifies preservation and modification regions $R_{pres}$ and $R_{chg}$ and orders three instructions by increasing expected edit extent. The verifier evaluates minimal editing on a five-point scale, with $s_{min} \geq 4$ required for acceptance, together with $s_{es} \geq 4$ and $s_{pl} \geq 3$. This is a planning preference and a verifier-based quality constraint, rather than minimization of a pixel-distance objective or a guarantee of a globally minimal edit. Each instruction can fall back to full-image regeneration, as shown in Algorithm 1.

**Algorithm 1** Verified hallucination materialization in RealmForgePipe

**Require:** $(I, q, y^\star, y_m)$; verifier $\mathcal{V}$; editor $\mathcal{E}$ (`gpt-image-2`)
**Require:** $(\tau_{es}, \tau_{pl}, \tau_{min}) = (4, 3, 4)$; $R_{ins} = 3$; $R_{syn} = 2$
**Ensure:** Curated record $(I^\star, q, y^\star \to y_m, \text{metadata})$, or DROP/FAIL

**if** $I$ unavailable **then return** DROP
$\hat{y} \leftarrow \mathcal{V}(I, q)$ ▷ 1. Triage: neither supplied answer is shown
$(A, B, C, D, e) \leftarrow \textsc{Audit}(q, \hat{y}, y^\star, y_m)$
  $A$: well-defined question; $B$: $\hat{y} \equiv y^\star$; $C$: $y_m \equiv y^\star$; $D$: refusal; $e$: error type
**if** $\neg A \vee \neg B \vee C \vee D$ **then return** DROP
$(d, h) \leftarrow \textsc{AssessDifficulty}(\mathcal{V}, I, q, y_m, e)$ ▷ 1.5. Record only; never reject
  $d \in \{1, \ldots, 5\}$: editing/verification difficulty (higher is harder); $h$: optional difficulty tags
$(t_0, R_{pres}, R_{chg}, \langle i_1, i_2, i_3 \rangle, p_{t2i}) \leftarrow \textsc{Plan}(\mathcal{V}, I, q, y^\star, y_m, e)$ ▷ 2. Plan
$hard_case \leftarrow [t_0 = \text{T2IReplan} \wedge e = \textsc{PerceptionLimit}]$ ▷ 2.5. Record only
  $t_0$: local editing, global editing, or text-to-image replanning; $i_1, i_2, i_3$: increasing edit extent
$\mathcal{T} \leftarrow \text{Dedup}(\langle t_0, \text{T2IReplan} \rangle)$ ▷ Preserve fallback order
**for** $k = 1, \ldots, R_{ins}$ **do**
  **for** $t \in \mathcal{T}$ **do**
    **for** $a = 1, \ldots, R_{syn}$ **do**
      $I' \leftarrow \textsc{Synth}(\mathcal{E}, I, \text{Wrap}(i_k, e), t, p_{t2i})$ ▷ 3. Synthesis
      **if** $I' = \emptyset$ **then continue** ▷ Next synthesis attempt
      $(s_{es}, s_{pl}, s_{min}) \leftarrow \textsc{Verify}(\mathcal{V}, I, I', q, y_m, R_{pres})$ ▷ 4. Quality verification
      **if** $s_{es} < \tau_{es} \vee s_{pl} < \tau_{pl} \vee s_{min} < \tau_{min}$ **then break**
      $mode \leftarrow \textsc{Classify}(q, y_m) \in \{\text{strict}, \text{relaxed}, \text{skip}\}$ ▷ 4b. Blind-check gate
      **if** $mode =$ strict **then**
        $\tilde{y} \leftarrow \mathcal{V}(I', q)$ ▷ Target answer is hidden during answering
        $m \leftarrow \textsc{Equiv}(\mathcal{V}, \tilde{y}, y_m)$
      **else if** $mode =$ relaxed **then**
        $f \leftarrow \textsc{ExtractFeature}(\mathcal{V}, R_{chg}, y_m)$
        $m \leftarrow [\mathcal{V}(I', \text{“Is feature } f \text{ visibly present?”}) = \text{YES}]$
      **else**
        $m \leftarrow \text{true}$ ▷ Skip the additional check
      **end if**
      **if** $m$ **then**
        $r_{human} \leftarrow \textsc{HumanReview}(I, I', q, y^\star, y_m)$ ▷ Final annotation review
        **if** $\neg r_{human}$ **then return** DROP ▷ Exclude rejected edits
        $u \leftarrow \textsc{UploadOSS}(I', \texttt{images_modi/}, \text{basename}(I))$
        $meta \leftarrow \{i_k, t, s_{es}, s_{pl}, s_{min}, mode, d, h, hard_case, r_{human}\}$
        **return** $\textsc{Curate}(u, q, y^\star, y_m, meta)$ ▷ 5. Store record
      **end if**
      **if** $mode \neq$ strict **then break** ▷ Only strict failures retry this pair
    **end for**
  **end for**
**end for**
**return** FAIL

**Triage subroutine (Stage 1).** The verifier first answers the original question without access to either supplied answer; counting questions require counting objects individually. It then determines question validity ($A$), agreement with the ground-truth answer ($B$), equivalence between the target response and the ground-truth answer ($C$), refusal ($D$), and the error type $e$. In optional two-pass auditing, both passes must satisfy $A$ and $B$, while either pass triggering $C$ or $D$ causes rejection; the stricter audit judgment is retained.

**Blind-check subroutine (Stage 4b).** The algorithm explicitly distinguishes three modes. Strict mode obtains an answer from the edited image with the target response hidden, then judges factual equivalence to $y_m$. Relaxed mode extracts a target visual feature from $(R_{chg}, y_m)$ and accepts only a YES response to its presence in the edited image. Skip mode performs no additional check and returns true. Only strict-mode check failures retry the same instruction–type pair; failed synthesis also consumes an attempt and retries, whereas a quality-threshold failure or a relaxed-mode failure advances to the next editing type. All verification scores use a five-point scale.

## B Qualitative Examples of Hallucination Mining

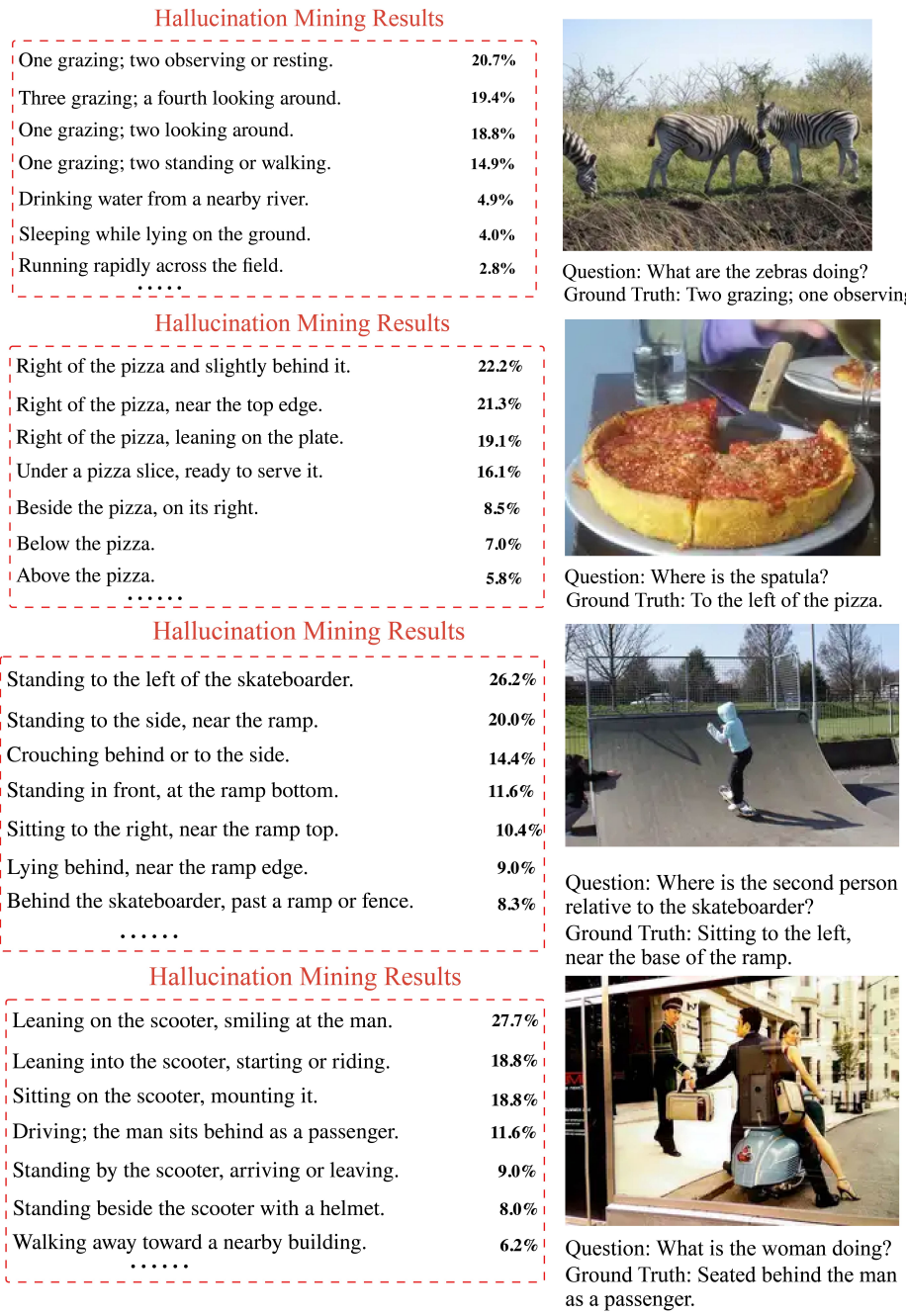



Figure 4: Hallucination mining results from RealmForgePipe on four examples involving actions and spatial relations. Each example shows the original image, question, ground-truth answer, and selected mined responses ranked by likelihood. Percentages denote candidate-relative weight $\rho_i(e)$ within the full error candidate pool, rather than calibrated answer probabilities; displayed values are not renormalized over the shown subset. Responses are shortened for readability.

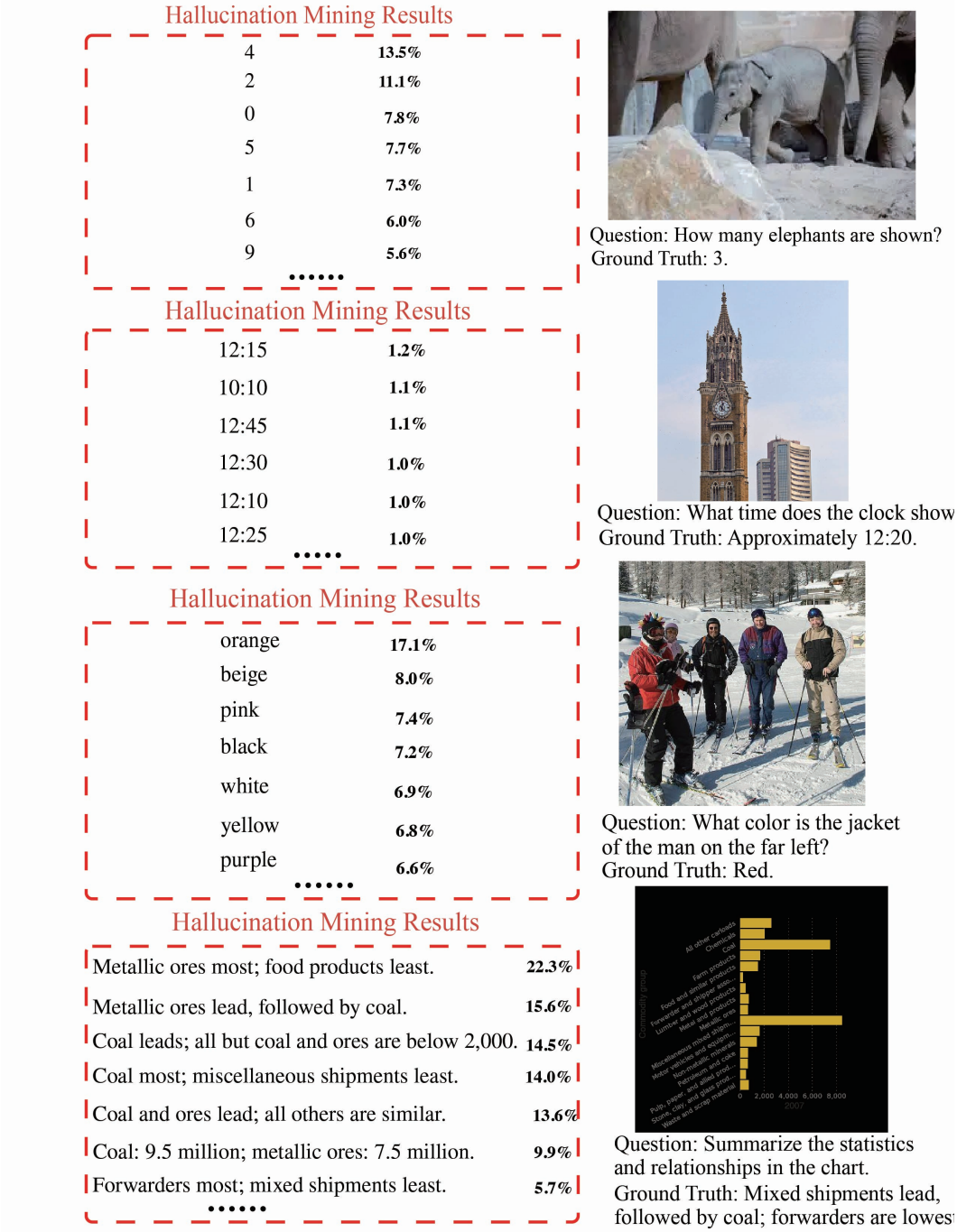



Figure 5: Additional hallucination mining examples covering counting, clock reading, color recognition, and chart analysis. Selected candidates are shown with their candidate-relative weights from the original mining runs, without renormalization over the displayed subset. These weights are not calibrated answer probabilities. Chart responses are shortened for readability.

## C HALLUCINATION-GUIDED IMAGE EDITING EXAMPLES

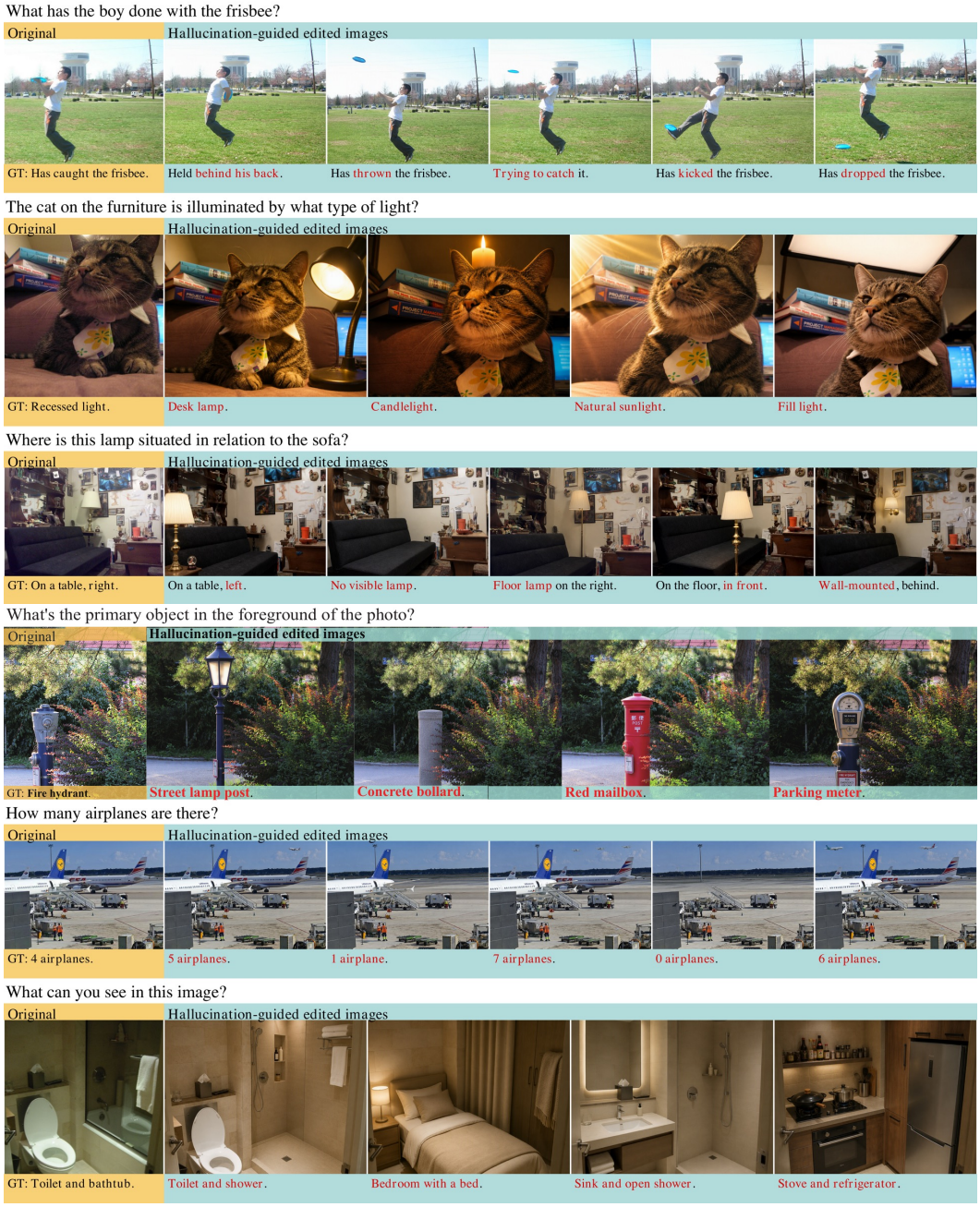



Figure 6: Hallucination-guided image editing examples from RealmForgePipe (1/2). From top to bottom: actions, illumination, spatial relations, mirror arrangements, object counts, and scene structure. Each row pairs the original image and ground-truth answer (yellow) with edited images and their corresponding hallucinated responses (cyan). Red text highlights the response attributes targeted by editing. Responses are shortened for readability.

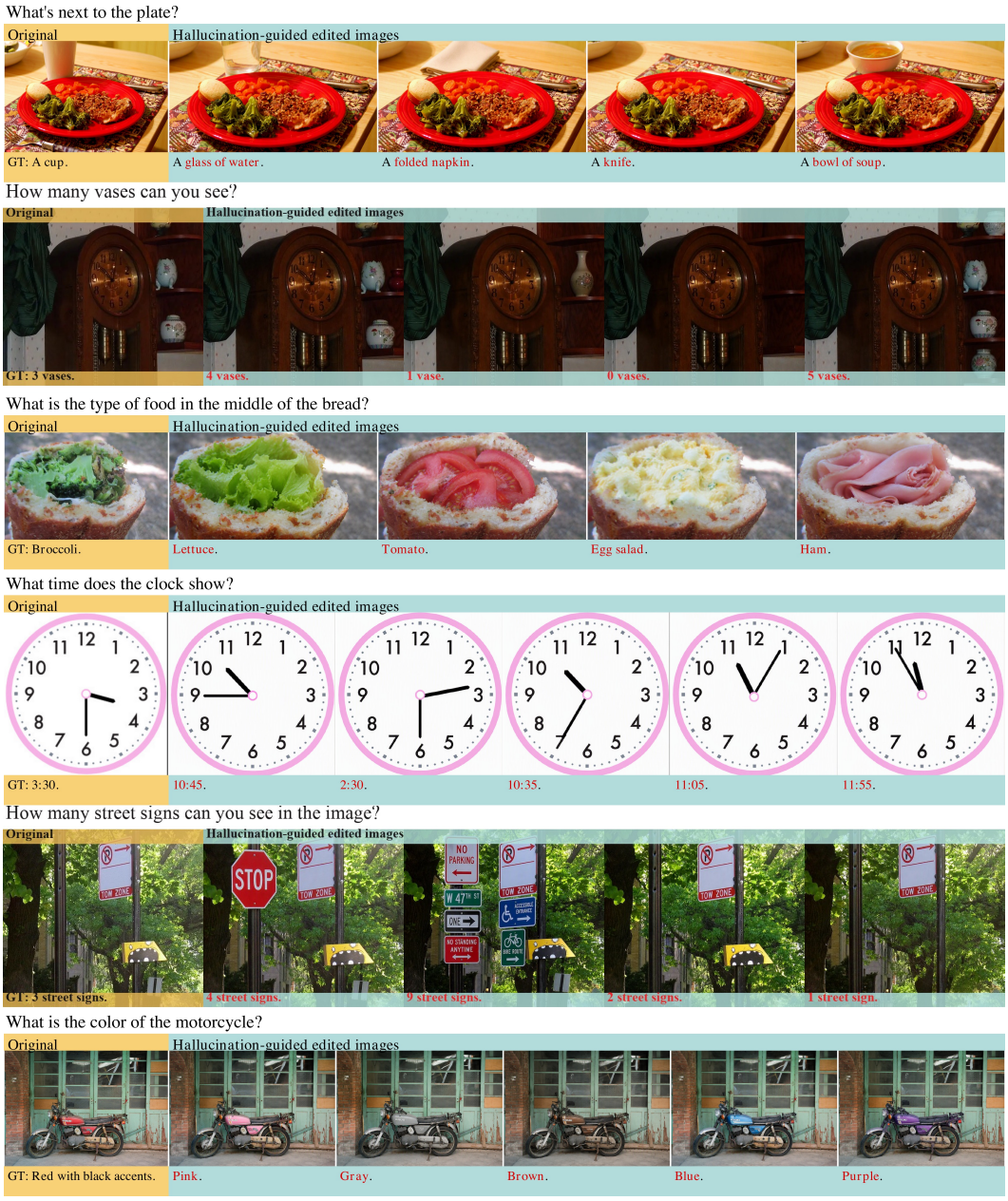



Figure 7: Hallucination-guided image editing examples from RealmForgePipe (2/2). From top to bottom: local object replacement, object–container relations, food content, clock time, surface patterns, and color attributes. Together with Figure 6, the two pages show 12 original-image families and 54 edited images. Yellow and cyan labels denote original and edited image–response pairs; red text highlights the targeted attributes.

# D Data Construction and Category-wise Statistics

**Source data and hallucination mining.** We construct our data from a subset of the VQA data released with VC-STaR (VisCoR-55K) (Pan et al., 2026), as described in Section 4. This subset provides 48,012 image–question inputs for RealmForgePipe. The retained and scored hallucination candidate pools contain 487,812 responses in total. Likelihood-based filtering and Top-$K_{\text{mine}}$ selection retain 114,982 responses (23.57% of the candidate pool) for subsequent processing. Table 4 summarizes the following construction stages from mined responses to human-approved edited pairs. These counts refer to retained candidate responses rather than raw sampling draws. Mining also explores latent hallucinations for inputs whose standard decoded response is correct; therefore, the number of mining inputs does not represent the number of observed hallucinations.

Table 4: Data construction summary for RealmForgePipe.

| Data | Count |
|---|---|
| Mined hallucinated responses | 114,982 |
| Candidates entering image editing | 96,320 |
| Successfully generated edited images | 91,986 |
| VLM blind-check approvals | 43,169 |
| Human-approved edited pairs | 28,960 |

**Overall editing yield.** Of the 114,982 selected candidates, 96,320 enter image editing and 28,960 yield human-approved edited pairs, giving an overall editing pass rate of 30.07%. Category sums and discrepancies are detailed in Appendix D.1.

**Final retained data.** The final retained HG-Edited Data contain 28,960 unique edited examples: 27,038 in the training set and 1,922 in the held-out test set. The training edits span 12,340 original-image families. No original-image family appears in both the training and test sets. The test set additionally includes 3,078 original examples and 1,922 edited examples, for 5,000 examples in total.

### D.1 Category-wise Editing and Review Statistics

Table 5 reports editing and review counts and empirical pass rates across 22 question categories. For each category $c$, we compute

$$\mathrm{PassRate}_c = \frac{N_{\mathrm{human},c}}{N_{\mathrm{synthesis},c}} \times 100\%,$$

where $N_{\mathrm{synthesis},c}$ is the number of candidates entering image editing and $N_{\mathrm{human},c}$ is the number of human-approved edited pairs. The denominator counts candidates rather than individual generation attempts. This measures the end-to-end yield of editing and review, rather than single-attempt generation success or the conditional acceptance rate among human-reviewed samples.

Table 5: Category-wise data construction counts and final pass rates. Synthesis denotes candidates entering image editing; Generated denotes successfully generated edited images; BlindCheck denotes records marked `BlindCheck OK`; Human denotes human-approved edited pairs. Final pass rate is Human / Synthesis.

| Category | Synthesis | Generated | BlindCheck | Human | Final pass rate (%) |
|---|---|---|---|---|---|
| count | 56,518 | 53,900 | 24,752 | 14,552 | 25.75 |
| other | 7,980 | 7,488 | 3,246 | 3,245 | 40.66 |
| count_by_attribute | 6,503 | 6,296 | 2,269 | 1,940 | 29.83 |
| shape | 5,450 | 5,298 | 2,008 | 1,227 | 22.51 |
| color | 5,831 | 5,657 | 2,763 | 2,429 | 41.66 |
| chart_describe | 2,705 | 2,633 | 2,027 | 1,242 | 45.91 |
| measurement | 2,397 | 2,337 | 1,097 | 729 | 30.41 |
| identification | 1,677 | 1,582 | 528 | 527 | 31.43 |
| text | 1,455 | 1,404 | 964 | 644 | 44.26 |
| size | 990 | 968 | 673 | 380 | 38.38 |
| position | 827 | 789 | 529 | 299 | 36.15 |
| material | 1,047 | 1,024 | 688 | 440 | 42.02 |
| scene_description | 764 | 687 | 499 | 498 | 65.18 |
| sentiment | 396 | 285 | 166 | 149 | 37.63 |
| existence | 387 | 355 | 256 | 216 | 55.81 |
| action | 450 | 398 | 149 | 148 | 32.89 |
| reasoning | 327 | 286 | 127 | 106 | 32.42 |
| time | 407 | 395 | 299 | 158 | 38.82 |
| probability | 123 | 120 | 83 | 19 | 15.45 |
| comparison | 31 | 31 | 8 | 7 | 22.58 |
| chart_read | 49 | 47 | 36 | 4 | 8.16 |
| multi_image | 6 | 6 | 2 | 1 | 16.67 |
| Total | 96,320 | 91,986 | 43,169 | 28,960 | 30.07 |

**Image editing configuration and cost.** We use GPT-Image-2 with the quality parameter set to `low`. The approximate image-generation cost is CNY 0.05 per generated image, excluding VLM verification and human annotation. This is a per-generation cost, not a cost per accepted edited pair: retries and generated images that fail subsequent checks also contribute to construction cost. The human annotation protocol and service price are described in Appendix D.2.

### D.2 Human Annotation Review Protocol

**Annotators and task.** Human review is conducted by 12 annotators, all from professional data annotation companies. Before annotation begins, the task and decision criteria are communicated to the annotators. Given an edited image and its proposed response, each annotator judges whether the response is supported by the image. Samples for which this judgment is uncertain are discarded. Each sample is initially reviewed by one annotator, and annotators work on separate, non-overlapping batches.

**Quality control.** A second quality check is performed on 10% of the data. If an annotator fails this quality check, the data assigned to that annotator are returned and reassigned to a different

annotator for reannotation. Only samples accepted through this review process are retained in the final HG-Edited Data.

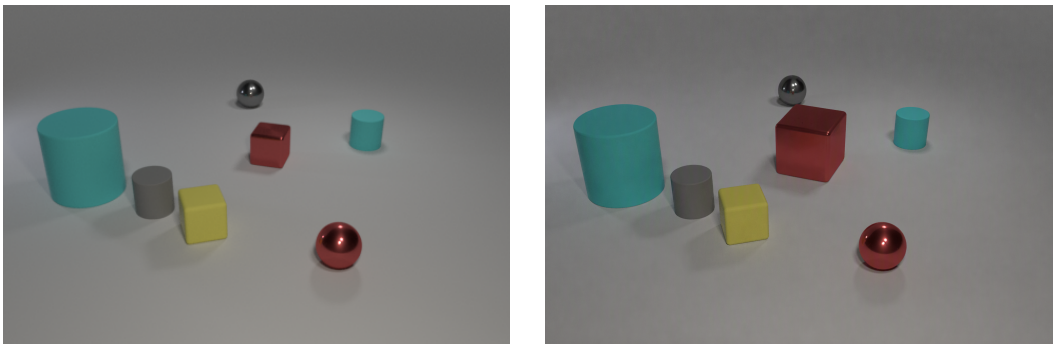
**Throughput and cost.** The review throughput is 500 samples per annotator-hour, and the annotation service is charged at CNY 0.0568 per sample.

## E LIMITATIONS

DreamRealm incurs data construction costs, including image generation with GPT-Image and human review, as detailed in Appendices D and D.2, respectively. In addition, DreamRealm is constrained by the capabilities of the underlying image-editing model, and some scenes remain difficult to edit successfully. Appendix D.1 (Table 5) reports the final pass rates of RealmForgePipe across 22 question categories, measured as the proportion of synthesis candidates ultimately approved by human review.

## F A WORKED EXAMPLE OF HALLUCINATION-GUIDED EDITING

We trace the recorded automated editing run for case `orig_148`, variant 0. The question is *What is the size of the red block?* The original answer is *Small*; the target hallucinated response is *Large*. This trace contains one target response and three alternative editing instructions for that response.



Original image: **Small** Edited image: **Large**

Figure 8: A recorded editing example. Enlarging the red cube makes the target response *Large* valid for the same question.

**Screening and difficulty assessment.** The verifier (`gemini-2.5-pro-06-17`; Gemini 2.5 (Comanici et al., 2025)) returns *Small*, marks the question and ground-truth answer as valid, and identifies an attribute error. It assigns difficulty 4/5 with shape-preservation, shadow-manipulation, and reflection-manipulation tags. These tags do not block editing.

**Planning.** The verifier selects local editing, identifies the red cube as the object to modify, and specifies that the background, lighting, and other objects should be preserved. It proposes three instructions:

1. Increase the size of the red cube.
2. Scale up the red cube to be approximately twice its current size.
3. Make the red cube a *large* object by increasing its size so its height is comparable to the large cyan cylinder on the left, and adjust its shadow accordingly.

**Execution and automated verification.** GPT-Image-2 (`gpt-image-2-0421-global`) executes the first instruction, prefixed with `[Error type: attribute]`. Generation takes 22.746 seconds. The verifier assigns 5/5 to each of edit success, visual plausibility, and minimal editing, exceeding the respective thresholds of 4, 3, and 4. These are the verifier's reported scores.

**Blind check and recorded outcome.** Given only the edited image and the question, the verifier answers that the red block is *large*. A subsequent equivalence check against *Large* returns `YES`. The run stores the edited image and metadata with status `ok`. The first generation under the first instruction succeeds, so neither alternative instruction nor the full-redraw fallback is executed. The supplied trace ends at automated storage and contains no human-review decision; final dataset acceptance remains subject to the human-review protocol in Appendix D.2.

## G TRAINING SETTINGS

**Training data and model configurations.** We use Qwen3.5-9B as both the hallucination-mining model and the primary training model. We train on original examples and the 27,038 HG-Edited training examples in RealmSet (Section 3.3). We partition the data by original-image family, keeping each original image and all its associated edits in the same split. We additionally reserve 5,000 in-distribution examples as a held-out test set, with no original-image family shared with the training data. All showcase examples in the appendix are drawn from this set. To examine the model specificity of hallucination mining, we also train Qwen2.5-VL-7B (Bai et al., 2025), Gemma-4-E4B-it (Gemma Team, 2026), and InternVL3.5-8B (Wang et al., 2025) separately on the same data, without additional hallucination mining for these target models.

**Model adaptation.** For Qwen3.5-9B, we initialize from the original pretrained checkpoint and optimize the RealmMatch objective in Section 3.4. We update the language model and multimodal projector while freezing the vision tower, without using LoRA adapters. This configuration has 8,996,576,512 trainable parameters. Training runs on eight NVIDIA H100 GPUs with BF16 precision, gradient checkpointing, and DeepSpeed ZeRO-3 with CPU optimizer offloading. Dropout is disabled. The training hyperparameters are summarized in Table 6.

Table 6: Training hyperparameters for the Qwen3.5-9B RealmMatch configuration. Batch units are complete question-centered families, rather than individual image–response combinations.

| Hyperparameter | Value |
|---|---|
| GPUs | 8 NVIDIA H100 |
| Training epochs | 3 |
| Peak learning rate | $5 \times 10^{-6}$ |
| Learning-rate schedule | Cosine |
| Warmup ratio | 0.03 |
| Weight decay | 0 |
| Maximum gradient norm | 1.0 |
| Per-device micro-batch | 1 family |
| Gradient accumulation | 10 micro-steps |
| Effective batch size | 80 families |
| Maximum sequence length | 6,144 tokens |
| Image processor pixel budget | 262,144 |
| Contrastive temperature $\tau$ | 1.0 |
| Row-wise contrast weight $\lambda_r$ | 0.5 |
| Column-wise contrast weight $\lambda_c$ | 0.5 |
| SFT loss coefficient | 1.0 |
| Precision | BF16 |
| Checkpoint saving | Every epoch |

**Family-wise computation.** Each training micro-step contains one complete family and uses the loss defined in Section 3.4: edited families use SFT and bidirectional contrast, whereas families without accepted edits use SFT and row-wise contrast. Matrix cells are processed in micro-batches of at most four cells and 24,576 padded input tokens. Families that fit in one micro-batch are backpropagated directly; otherwise, we first assemble the complete score matrix from detached scores and then replay the cell micro-batches using gradients derived from the full family loss. This procedure preserves the full family-level gradient while retaining activations for only one cell micro-batch at a time. The implementation uses PyTorch (Paszke et al., 2019) 2.10.0, Transformers (Wolf et al., 2020) 5.3.0, Accelerate 1.11.0, and DeepSpeed 0.18.4.

# H EVALUATION

## H.1 BENCHMARK SCORING

**Benchmarks and metrics.** We evaluate fine-grained visual perception and hallucination on six benchmarks: HallusionBench (Guan et al., 2024), MMVP (Tong et al., 2024b), AMBER (Wang et al., 2023), MMHal-Bench (Sun et al., 2024), CV-Bench-2D (Tong et al., 2024a), and ObjectHal (Rohrbach et al., 2018). We report aAcc for HallusionBench, item and pair accuracy for MMVP, average accuracy for AMBER, hallucination rate for MMHal-Bench, macro accuracy for CV-Bench-2D, and CHAIRs for ObjectHal. Lower hallucination rate and CHAIRs indicate better performance; higher values are better for the other metrics. We also provide model inference results in the supporting material.

Where a benchmark calls an LLM judge we use `Qwen3.8-flash` at temperature 0, with thinking off and no image in the judge context. VLMEvalKit protocols follow the toolkit release (Duan et al., 2024).

**MMVP.** Under VLMEvalKit's MMVP protocol, a binary hit $h_i \in \{0, 1\}$ marks whether item $i$ matches the gold choice ($N = 300$). Pair accuracy further requires both members of a consecutive pair to be right (150 pairs):

$$\mathrm{ItemAcc} = \frac{1}{N}\sum_{i=1}^{N} h_i, \qquad \mathrm{PairAcc} = \frac{1}{N/2}\sum_{p=1}^{N/2} h_{2p-1}\, h_{2p}.$$

**HallusionBench.** Scoring follows the authors' GPT check on all $N = 1{,}129$ questions: the judge labels a prediction correct, incorrect, or unclear against the detailed reference. Unclear is credited on the 178 imageless visual-supplement items. The reported figure is item-level accuracy

$$\mathrm{aAcc} = \frac{1}{N}\sum_{i=1}^{N} h_i.$$

**AMBER.** VLMEvalKit's AMBER rating maps replies to Yes/No (the judge is invoked only when the label is unresolved), then averages Attribute, Existence, and Relation accuracy with equal weight:

$$\mathrm{AvgACC} = \frac{1}{3}\big(\mathrm{Acc}_{\mathrm{attr}} + \mathrm{Acc}_{\mathrm{exist}} + \mathrm{Acc}_{\mathrm{rel}}\big).$$

**MMHal-Bench.** The original five-shot MMHal rubric assigns each of $N = 96$ answers a score $s_i \in \{0, \ldots, 6\}$. The reported hallucination rate is the fraction of answers below 3:

$$\mathrm{HalRate} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[s_i < 3].$$

**CV-Bench-2D.** VLMEvalKit's CV-Bench scorer reports 2D accuracy as a macro-average over sources ($N = 1{,}438$). If source $c$ has accuracy $\mathrm{Acc}_c$,

$$\mathrm{Acc}_{\mathrm{2D}} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \mathrm{Acc}_c.$$

**ObjectHalBench.** Object mentions are extracted and matched to COCO with the RLHF-V CHAIR pipeline. A mention is hallucinated when it does not match any annotated object in the image. With $N = 300$ responses, $h_i \in \{0, 1\}$ is 1 if response $i$ contains at least one such mention:

$$\mathrm{CHAIR}_s = \frac{1}{N}\sum_{i=1}^{N} h_i.$$

### H.2 IN-DISTRIBUTION EVALUATION

In addition to the six public benchmarks, we evaluate the models on a held-out in-distribution split of RealmSet containing 5,000 examples. The split consists of 3078 original examples, 1922 HG-Edited examples. Table 7 reports the results on the held-out RealmSet test set.

Table 7: In-distribution accuracy on the 5,000-example held-out test set.

| Model / Method | Overall ↑ | Original ↑ | HG-Edited ↑ |
|---|---|---|---|
| Qwen3.5-9B (Raw) | 64.80 | 66.44 | 62.17 |
| + DreamRealm | **68.30** | **70.76** | **64.36** |
| Qwen2.5-VL-7B (Raw) | 48.40 | 51.23 | 43.86 |
| + DreamRealm | 59.66 | 61.99 | 55.93 |
| Gemma-4-E4B-it (Raw) | 39.54 | 40.90 | 37.36 |
| + DreamRealm | 50.54 | 53.09 | 46.46 |
| InternVL3.5-8B (Raw) | 54.64 | 55.88 | 52.65 |
| + DreamRealm | 60.16 | 62.44 | 56.50 |

# I QUALITATIVE COMPARISONS WITH THE BASE MODEL

Figures 9–10 show 32 selected comparisons across diverse visual tasks. Each page contains 16 cases. Questions and answers are condensed while preserving their meaning; choice letters are expanded where useful. Ref. denotes the supplied reference; Qwen35-9b (DreamRealm) and Qwen35-9b (Raw) denote the adapted and original models, respectively. These selected examples do not estimate the overall success rate.

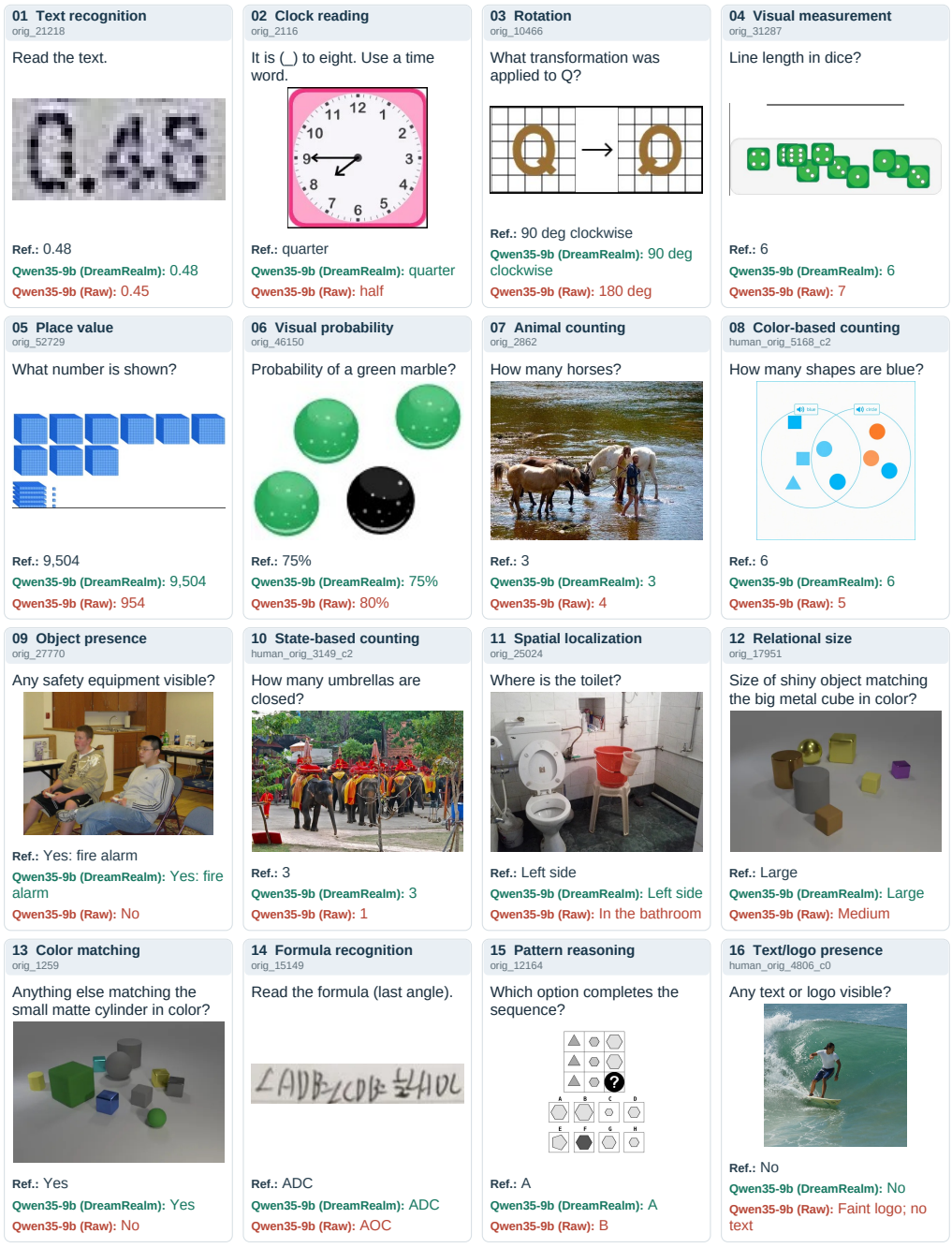



Figure 9: Compact qualitative comparisons (1/2), cases 1–16. Green and red identify model answers, not independent correctness judgments. Case 14 shows only the differing final angle in the formula. IDs beginning with `human_` denote edited inputs.

**17 Text recognition**
orig_4879

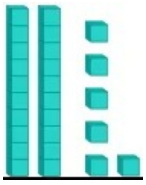
Read the text.




**Ref.:** 7.95
**Qwen35-9b (DreamRealm):** 7.95
**Qwen35-9b (Raw):** 795

**18 Place value**
orig_37421

What number is shown?



**Ref.:** 26
**Qwen35-9b (DreamRealm):** 26
**Qwen35-9b (Raw):** 25

**19 Object counting**
orig_11450

How many propellers does the plane have?



**Ref.:** 2
**Qwen35-9b (DreamRealm):** 2
**Qwen35-9b (Raw):** 4

**20 People counting**
orig_33362

How many people on the bike?



**Ref.:** 2
**Qwen35-9b (DreamRealm):** 2
**Qwen35-9b (Raw):** 1

**21 Object counting**
orig_49162

How many forks on the table?



**Ref.:** 1
**Qwen35-9b (DreamRealm):** 1
**Qwen35-9b (Raw):** 2

**22 Reflection counting**
orig_28677

How many cars are reflected in the mirror?



**Ref.:** 2
**Qwen35-9b (DreamRealm):** 2
**Qwen35-9b (Raw):** 1

**23 Object counting**
orig_22305

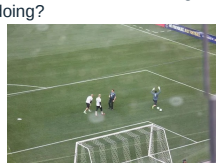
How many garlic bulbs?



**Ref.:** 2
**Qwen35-9b (DreamRealm):** 2
**Qwen35-9b (Raw):** 1

**24 Action recognition**
orig_19101

What is the man on the right doing?



**Ref.:** Stretching
**Qwen35-9b (DreamRealm):** Stretching
**Qwen35-9b (Raw):** Kicking

**25 Object counting**
orig_34461

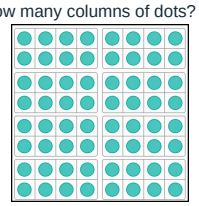
How many cupcakes?



**Ref.:** 5
**Qwen35-9b (DreamRealm):** 5
**Qwen35-9b (Raw):** 6

**26 Array structure**
human_orig_6582_c4

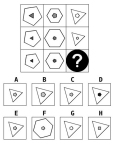
How many columns of dots?



**Ref.:** 8
**Qwen35-9b (DreamRealm):** 8
**Qwen35-9b (Raw):** 4

**27 Pattern reasoning**
orig_13460

Which option completes the sequence?



**Ref.:** C
**Qwen35-9b (DreamRealm):** C
**Qwen35-9b (Raw):** B

**28 Object counting**
orig_40094

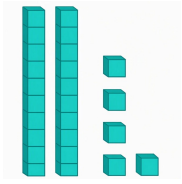
How many windows on the building?




**Ref.:** 4
**Qwen35-9b (DreamRealm):** 4
**Qwen35-9b (Raw):** 3

**29 Place value**
human_orig_37421_c0

What number is shown?



**Ref.:** 25
**Qwen35-9b (DreamRealm):** 25
**Qwen35-9b (Raw):** 24

**30 Sign counting**
human_orig_33746_c1

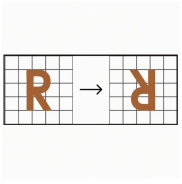
How many one-way signs?



**Ref.:** 4
**Qwen35-9b (DreamRealm):** 4
**Qwen35-9b (Raw):** 3

**31 Transformation**
human_orig_6811_c0

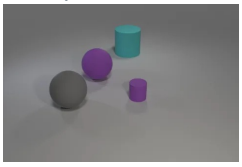
What happened to the letter?



**Ref.:** Turn
**Qwen35-9b (DreamRealm):** Turn
**Qwen35-9b (Raw):** Flip

**32 Relational size**
orig_6081

Size of the ball matching the small object in color?



**Ref.:** Large
**Qwen35-9b (DreamRealm):** Large
**Qwen35-9b (Raw):** Medium

Figure 10: Compact qualitative comparisons (2/2), cases 17–32. Reflected vehicles in case 22 include the car carrying the dog. All responses are condensed from the recorded outputs.

## J DETAILS OF COMPARED METHODS

Block I of Table 1 reports specialized hallucination-mitigation methods. Published numbers for these methods are not directly comparable: prior work uses different judges and scoring rules. We therefore re-evaluate every Block I method under the single protocol in Appendix H.1 and do not copy scores from the original papers. When a public checkpoint is available, we load that release and only re-run inference and scoring. Otherwise we follow the published training recipe on the backbone in Table 8, then score the resulting model with the same pipeline. The methods were proposed on different vision–language backbones, so Block I is a protocol-controlled comparison of released recipes rather than a same-backbone ranking. Same-backbone comparisons are in Blocks II–III.

Table 8: Backbone and setup for each Block I method.

| Method | Backbone | Setup |
|---|---|---|
| VCD (Leng et al., 2024) | LLaVA-1.5-7B | Inference-time decoding |
| CoCI-CAP (Manevich and Tsarfaty, 2025) | LLaVA-OneVision (Qwen2-7B) | Inference-time decoding |
| RLAIF-V (Yu et al., 2025) | LLaVA-1.5-7B | Public checkpoint |
| SymMPO (Liu et al., 2026) | LLaVA-1.5-7B | Public checkpoint |
| mDPO (Wang et al., 2024a) | Bunny (Llama-3-8B) | Reproduced |
| VC-STaR (Pan et al., 2026) | Qwen2.5-VL-7B | Reproduced |
| MED (Bai et al., 2026) | Qwen2.5-VL-7B | Reproduced |
| IC-VCO (Deng et al., 2026) | LLaVA-OneVision (Qwen2-7B) | Reproduced |
| FINER (Xiao et al., 2026) | Qwen2.5-VL-7B | Public checkpoint |
| IRIS (Li et al., 2026) | LLaVA-1.5-7B | Reproduced |
| $C^2$-DPO (Ko et al., 2026) | LLaVA-1.5-7B | Reproduced |