Reliability-Prioritized Fine-Grained Generation in Multimodal Large Language Models

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs) are increasingly expected to generate fine-grained descriptions of visual content. However, we observe and theoretically show that generating fine-grained responses poses a reliability challenge, *i.e.*, fine-grained generation is more error-prone than coarse-grained generation. This phenomenon suggests that models should generate the finest description that remains reliable rather than simply produce more specific outputs. To investigate this problem, we develop GRANFACT, a granularity-aware benchmark consisting of expert-verified multi-object images with coarse-to-fine category annotations. Then, we design a hierarchy-aware evaluation algorithm, which assesses both whether model predictions are visually correct and how specific the correct predictions are. We also propose a reliability-prioritized preference optimization method based on Direct Preference Optimization, which penalizes unreliable fine-grained claims while rewarding reliable specificity. Experiments on GRANFACT show that our method improves fine-grained generation while preserving reliability.

1 Introduction

Multimodal large language models (MLLMs) have demonstrated strong visual understanding capabilities (Alayrac et al., 2022; Li et al., 2023a; Liu et al., 2023; Zhu et al., 2024; Dai et al., 2023), enabling open-ended descriptions of visual content across diverse scenarios. However, their reliability remains a significant challenge when generating fine-grained visual descriptions (Kim and Ji, 2024; He et al., 2025; Ye et al., 2025; Yu et al., 2025). We observe and theoretically show that producing a fine-grained generation is substantially more error-prone than producing a coarse-grained one for MLLMs. As illustrated in Figure 1, given an image of an "iPhone 17 Pro", a model can often correctly describe it as a "smartphone" or an "iPhone", yet when asked for a fine-grained description, it may misidentify it as a visually similar model, such as an "iPhone 17 standard".

This phenomenon suggests that fine-grained visual description should not simply encourage more specific responses. Instead, models should generate the finest reliable description, and back off to a coarser but correct one when finer-grained predictions become unreliable. We refer to this problem as reliability-prioritized fine-grained visual description generation, where the model first ensures correctness and then chooses the finest description among reliable candidates.

Existing visual description benchmarks (Dong et al., 2024; Cheng et al., 2025; Liu et al., 2025) and fine-grained recognition benchmarks (Wah et al., 2011; Krause et al., 2013; Maji et al., 2013; Kim and Ji, 2024; He et al., 2025; Yu et al., 2026) are insufficient for evaluating this capability. From the data perspective, existing benchmarks often focus on single objects or scenes with limited visual confusion, whereas realistic multi-object fine-grained benchmarks require substantial effort in image collection and coarse-to-fine annotation. From the evaluation perspective, existing protocols usually judge correctness at a predefined granularity, making it non-trivial to determine whether coarser or finer descriptions remain correct. As a result, they fail to reliably evaluate models across granularities. This issue is more challenging in open-ended multi-object generation, where evaluation must first identify which ground-truth (GT) object each prediction refers to, and then decide its granularity. Without global alignment, repeated or unsupported predictions may be incorrectly rewarded, while missed objects are overlooked.

To address these challenges, we construct a granularity-aware benchmark, namely GRANFACT. GRANFACT consists of 581 images across 7 domains. The images are collected from the Internet and real-world photographs with multiple visually

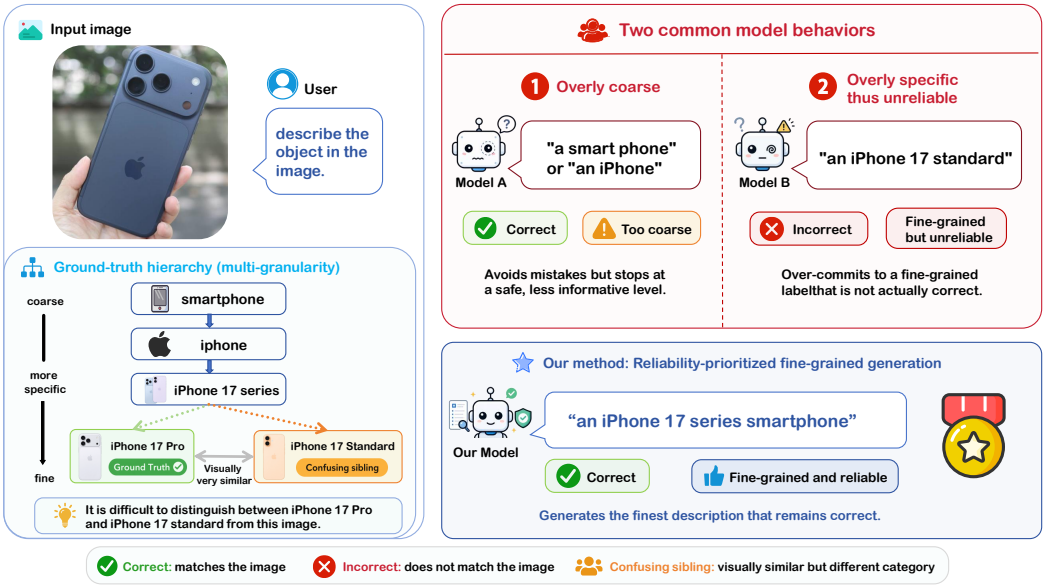

Figure 1: Motivation of reliability-prioritized fine-grained generation. Rather than simply maximizing specificity, a model should generate the finest description that remains reliable.

relevant objects. For each object, human experts provide category annotations at coarse-to-fine semantic granularities. For evaluation, we introduce a hierarchy-aware evaluation algorithm that parses open-ended responses into structured entities and assigns them to GT entities under quantity constraints. To resolve the assignments between predicted and GT entities, we introduce a reliability-prioritized objective that first prioritizes category-consistent or category-coarser assignments and then favors the finest-grained assignment. Based on the final assignment, we design two types of metrics. Granularity-neutral metrics count a prediction as correct if it matches any annotated category level, while granularity-aware metrics further measure the specificity of correct descriptions.

Based on Direct Preference Optimization (DPO) (Rafailov et al., 2023), we propose Reliability-Prioritized DPO (RP-DPO) to improve fine-grained generation of MLLMs under a reliability-first constraint. RP-DPO first applies rollback to hallucinated predicted objects by replacing them with coarse-grained ancestors shared with the GT. It then constructs reliability preferences between original and rolled-back responses, and granularity preferences among reliable responses. With larger margins for reliability, RP-DPO enforces reliability-first optimization while favoring finer-grained descriptions under comparable reliability. Experiments with Qwen3-VL-8B on GranFact highlight the importance of reliability-first optimization for fine-grained multimodal understanding.

In summary, our contributions are threefold:

- We introduce GranFact, a granularity-aware multimodal benchmark that includes expert-verified multi-object images with coarse-to-fine hierarchical labels.
- We propose a hierarchy-aware evaluation algorithm that aligns open-ended predictions with GT entities and measures the granularity and correctness of model responses.
- We propose RP-DPO, a reliability-prioritized preference optimization method that improves fine-grained visual description generation while preserving reliability.

2 Related Work

2.1 MLLM Benchmarks

Recent MLLM benchmarks evaluate broad multimodal abilities, including perception and cognition (Fu et al., 2025a), general multimodal capabilities (Liu et al., 2024; Zhang et al., 2025), visual reasoning (Lu et al., 2024), and expert-level understanding (Yue et al., 2024). More targeted benchmarks study fine-grained recognition (Yu et al., 2026) and hallucination detection (Li et al., 2023b; Wang et al., 2023; Guan et al., 2024; Cai et al., 2025; Yin et al., 2026). These benchmarks typically target fixed-granularity answers or binary correctness. In contrast, our benchmark jointly evaluates the correctness and granularity of open-ended descriptions.

2.2 Fine-Grained Visual Description

Prior work improves the visual specificity of MLLMs through detailed caption data (Chen et al., 2024), grounded generation (You et al., 2024; Rasheed et al., 2024), and fine-grained recognition training (He et al., 2025, 2026; Wu et al., 2026). These methods make outputs more detailed or discriminative, but do not explicitly prioritize reliability when increasing semantic specificity. We instead focus on reliability-prioritized fine-grained generation, encouraging more specific descriptions while preserving reliability.

2.3 Hallucination Mitigation in MLLMs

Hallucination mitigation aims to reduce visual claims that are not grounded in the input image. Existing methods address this issue through inference-time interventions, such as contrastive decoding (Leng et al., 2024; Tong et al., 2025; Chen et al., 2025), post-hoc correction (Zhou et al., 2024a; Yin et al., 2024), or additional supervision and alignment signals (Yu et al., 2024). These methods improve reliability by reducing ungrounded claims, but do not explicitly consider the granularity of grounded descriptions. Our work treats hallucination reduction as a prerequisite and further encourages finer-grained descriptions.

2.4 Direct Preference Optimization

Direct Preference Optimization (DPO) directly optimizes preference pairs without training an explicit reward model (Rafailov et al., 2023). Recent preference-optimization methods introduce target or adaptive margins to model different preference strengths (Meng et al., 2024; Wu et al., 2025; Sun et al., 2025). Other works extend DPO to multi-objective alignment (Zhou et al., 2024b) or multimodal settings (Wang et al., 2024; Fu et al., 2025b). Our method follows this line but focuses on reliability-prioritized fine-grained generation.

3 Analysis

We analyze reliability–granularity trade-off from both theoretical and empirical perspectives.

3.1 Theoretical Analysis

Let $\mathscr{S}$ be a semantic tree, where descendant nodes represent more fine-grained concepts. A semantic granularity is represented by a cut set, or tree cut (Wu et al., 2024), Ω, which defines the prediction space at that granularity. For each cut Ω, we define $\varepsilon^\star(\Omega)$ as the Bayes-optimal semantic risk under that granularity.

Theorem 1 (Monotonicity of Optimal Semantic Risk)**.** *Let Ω_f and Ω_c be two cut sets of $\mathscr{S}$, where Ω_f is finer than Ω_c. Then, it holds that*

$$\varepsilon^\star(\Omega_c) \leq \varepsilon^\star(\Omega_f). \tag{1}$$

The formal definition and proof are provided in Appendix A. Theorem 1 formalizes the reliability–granularity trade-off in a hierarchical semantic space, showing that generating at a finer granularity is inherently more error-prone.

3.2 Reliability–Granularity Trade-off

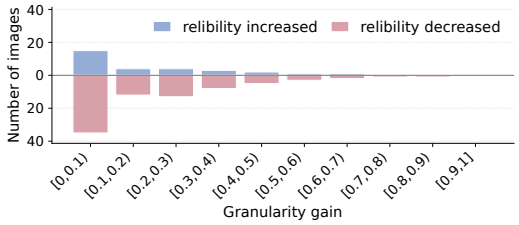

Figure 2: Reliability changes from conservative to aggressive prompting across per-image granularity gains.

We conduct an empirical analysis to examine the trade-off between granularity and reliability. We prompt a model to describe the same input images using either a conservative prompt or an aggressive prompt that asks for maximal detail. We then measure how reliability changes, as shown in Figure 2. Finer-grained generations are more frequently associated with reliability drops. This suggests that simply eliciting more specific descriptions is insufficient. Instead, models should generate finer-grained descriptions only when the additional specificity can be stated reliably.

4 GranFact

GranFact consists of a manually annotated fine-grained image dataset and an evaluation protocol.

4.1 Evaluation Set Construction

GranFact contains 581 images collected through web sourcing and real-world photography across 7 visual domains: daily objects, plants, animals, cars, electronics, landmarks, and games. Each annotated entity is manually labeled with multi-granularity categories, quantity, and visual attributes, with examples shown in Figure 3. Figure 4 summarizes granularity depth, annotated objects per image, and domain distribution, with further details provided in Appendix B.

Let $\mathcal{Z}$ denote the evaluation set. Each image is annotated with a set of GT entities $\mathcal{G} = \{g_j\}_{j=1}^n$. Each entity g_j is labeled with a coarse-to-fine category $\mathcal{H}_j = (c_{j,1}, \ldots, c_{j,L_j})$, a quantity capacity d_j,

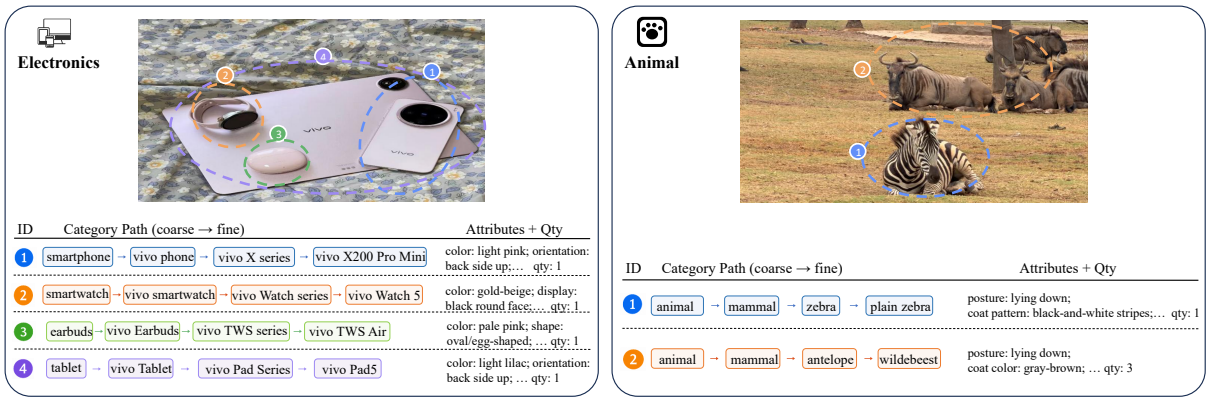

Figure 3: Qualitative examples of dataset annotations across different domains.

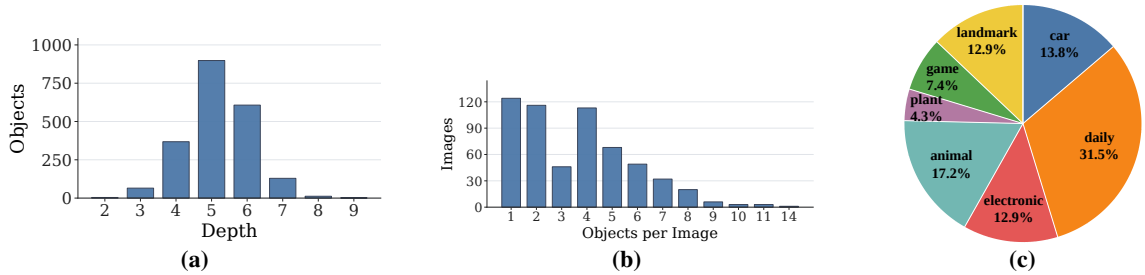

Figure 4: Dataset statistics of GRANFACT. (a) Distribution of max granularity depth per object. (b) Distribution of annotated objects per image. (c) Domain distribution of images.

and visual attributes, where $c_{j,1}$ and c_{j,L_j} denote the coarsest and finest categories, respectively.

4.2 Hierarchy-Aware Evaluation Algorithm

We propose a hierarchy-aware evaluation pipeline for open-ended visual descriptions. It first constructs a feasible set of prediction–GT assignments, then selects the final assignment with a reliability-prioritized, granularity-aware objective.

4.2.1 Response Parsing

Given an image and its corresponding MLLM response, we use an LLM to parse the free-form response into structured prediction entities. The parsed prediction set is denoted as $\mathcal{P} = \{p_i\}_{i=1}^{m}$, where each prediction entity p_i consists of a predicted category q_i, a predicted quantity s_i, and a set of non-quantitative attributes $\mathbf{u}_i$. The full parsing details are provided in Appendix C.1.

4.2.2 Prediction–GT Assignment

After parsing an open-ended response into structured prediction entities, evaluation requires assigning predicted entities to annotated GT entities. This assignment is non-trivial because the number of predicted entities and GT entities may differ, while model outputs can appear at arbitrary semantic granularities and may include hallucinated entities.

We first augment the GT entity set with a dummy hallucination entity g_{n+1}, obtaining $\widetilde{\mathcal{G}} = \mathcal{G} \cup \{g_{n+1}\}$. We then define an assignment matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{m\times(n+1)}$ to represent the quantity assignment from prediction entities to the augmented GT entity set. Unlike a binary matching matrix, each entry in $\mathbf{X}$ specifies how much of a predicted quantity is assigned to a GT entity. For example, a prediction with quantity larger than one may be distributed across several compatible GT entities.

We constrain that an assignment matrix $\boldsymbol{X}$ must assign each predicted quantity to GT entities or the hallucination entity, *i.e.*, $\sum_{j=1}^{n+1} x_{ij} = s_i$. Moreover, we also constrain the total quantity assigned to each GT entity g_j to be at most its annotated capacity d_j, *i.e.*, $\sum_{i=1}^{m} x_{ij} \leq d_j$. The hallucination entity g_{n+1} is left unconstrained, so unsupported or over-counted predictions can always be assigned to it and counted as hallucination. Overall, the constraints on $\boldsymbol{X}$ can be summarized as

$$\mathcal{X} = \Big\{\mathbf{X} \in \mathbb{R}_{\geq 0}^{m\times(n+1)} \,\Big|\, \sum_{j=1}^{n+1} x_{ij} = s_i, \quad 1 \leq i \leq m, \quad \sum_{i=1}^{m} x_{ij} \leq d_j, \quad 1 \leq j \leq n\Big\}. \tag{2}$$

4.2.3 Objective of Prediction–GT Assignment

With the assignment set $\mathcal{X}$, we select the final assignment matrix using a reliability-prioritized

objective. This objective is based on two assignment scores, *i.e.*, a reliability score $M(\mathbf{X})$ and a granularity-aware score $M_{\text{gran}}(\mathbf{X})$. $M(\mathbf{X})$ measures the degree to which predicted entities are assigned to category-compatible GT entities, where an assignment is considered category-compatible if the predicted category matches the GT category or one of its ancestor categories. $M_{\text{gran}}(\mathbf{X})$ further accounts for category granularity and attribute consistency. The final assignment matrix maximizes these two scores in lexicographic order, giving primary priority to $M(\mathbf{X})$ and using $M_{\text{gran}}(\mathbf{X})$ to distinguish assignments with the same reliability score, which can be modeled as[1]

$$\begin{aligned} \mathbf{X}^\star &\in \arg\max_{\mathbf{X}' \in \mathcal{X}} \quad M_{\text{gran}}(\mathbf{X}') \\ \text{s.t.} \quad \mathbf{X}' &\in \arg\max_{\mathbf{X} \in \mathcal{X}} M(\mathbf{X}). \end{aligned} \tag{3}$$

To compute the two assignment-level scores $M(\cdot)$ and $M_{\text{gran}}(\cdot)$, we define a granularity-aware assignment score w_{ij} for each prediction entity p_i and GT entity g_j. w_{ij} assigns positive scores to category-compatible assignments, with higher scores for finer category levels and higher attribute consistency. It assigns negative scores to incompatible assignments and neutral scores to assignments to the dummy GT entity. Formally, w_{ij} is computed as

$$w_{ij} = \begin{cases} \dfrac{\ell_{ij} + a_{ij}}{L_j + 1}, & j \le n,\ \ell_{ij} > 0, \\ -1, & j \le n,\ \ell_{ij} = 0, \\ 0, & j = n+1. \end{cases} \tag{4}$$

$\ell_{ij} \in \{0, \dots, L_j\}$ is the category granularity level assigned to (p_i, g_j), where $\ell_{ij} \le 0$ indicates category incompatibility and a larger value indicates a finer supported category level. $a_{ij} \in [0, 1]$ is the attribute consistency score between p_i and g_j, computed over their non-quantitative attributes. Using these scores, $M(\mathbf{X})$ and $M_{\text{gran}}(\mathbf{X})$ are computed as

$$\begin{aligned} M(\mathbf{X}) &= \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{I}[w_{ij} > 0] x_{ij}, \\ M_{\text{gran}}(\mathbf{X}) &= \sum_{i=1}^{m} \sum_{j=1}^{n+1} w_{ij} x_{ij}. \end{aligned} \tag{5}$$

We solve the lexicographic assignment in Eq. (3) by casting the quantity assignment as a minimum-cost flow problem on a bipartite network (Ahuja et al., 1988) and solving it with a successive shortest augmenting-path algorithm on the residual network (Klein, 1967). Appendix C provides additional details on the overall evaluation algorithm.

[1] Optimal assignments may be non-unique but yield identical metric values; see Appendix C.4.

4.3 Evaluation Metrics

We compute metrics by aggregating quantities at two levels. For each example z, we aggregate over entities to obtain $S_z = \sum_i s_i$, $D_z = \sum_j d_j$, $M_z = M(\mathbf{X}_z^\star)$, and $M_{\text{gran},z} = M_{\text{gran}}(\mathbf{X}_z^\star)$. We omit the subscript z for dataset-level sums over examples, e.g., $S = \sum_{z \in \mathcal{Z}} S_z$ and similarly for D, M, and M_{gran}. GRANFACT reports granularity-neutral metrics and granularity-aware metrics.

Granularity-neutral metrics. Granularity-neutral metrics measure whether predictions are assigned to category-compatible GT entities, regardless of how fine-grained the descriptions are. The precision, recall, and F1 are

$$\mathrm{P} = \frac{M}{S}, \quad \mathrm{R} = \frac{M}{D}, \quad \mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}. \tag{6}$$

We also report the Image Reliability Rate that is

$$\mathrm{IR} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \mathbb{I}[M_z = S_z], \tag{7}$$

where an example is reliable only when all predicted quantities are assigned to category-compatible GT entities.

Granularity-aware metrics. Granularity-aware metrics further account for the specificity and attribute consistency of reliable predictions. The granularity-weighted precision, recall, and F1 are

$$\begin{aligned} \mathrm{P}_{\text{gran}} = \frac{M_{\text{gran}}}{S}, \quad \mathrm{R}_{\text{gran}} = \frac{M_{\text{gran}}}{D}, \\ \mathrm{F1}_{\text{gran}} = \frac{2\mathrm{P}_{\text{gran}}\mathrm{R}_{\text{gran}}}{\mathrm{P}_{\text{gran}} + \mathrm{R}_{\text{gran}}}. \end{aligned} \tag{8}$$

Finally, we report the granularity-aware counterpart of image reliability:

$$\mathrm{GIR} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \mathbb{I}[M_z = S_z] \cdot \frac{M_{\text{gran},z}}{M_z}, \tag{9}$$

where the granularity term is set to zero when $M_z = 0$. GIR gives non-zero credit only to fully reliable responses and weights them by their average granularity-aware score. As an additional diagnostic, we also report the average granularity score of reliable predictions

$$\mathrm{G}_{\text{avg}} = \frac{M_{\text{gran}}}{M}, \tag{10}$$

where G_{avg} is set to 0 when $M = 0$.

5 Reliability-prioritized DPO

We propose Reliability-prioritized DPO (RP-DPO), a preference optimization method for fine-grained visual description generation. It consists of reliability-guided semantic rollback (RSR), preference construction, and metric-margin DPO.

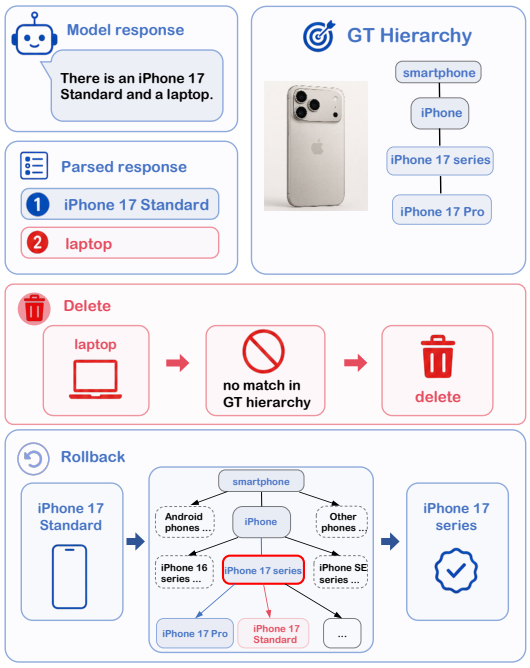

Figure 5: Illustration of RSR with DELETE and ROLLBACK operations.

5.1 Reliability-Guided Semantic Rollback

Since an image often contains multiple objects, models may fail to produce correct descriptions for all objects in responses. To construct reliable positive samples for DPO, RSR converts erroneous responses into semantically valid ones by rolling incorrect predictions back to the coarse-grained ancestor category shared with GT objects.

Specifically, we first parse the model response y_{orig} into prediction objects. For a hallucinated predicted object, we apply DELETE or ROLLBACK to correct it. If a predicted object and its ancestor categories have no intersection with the hierarchical annotations of any GT entity, RSR assigns a DELETE to remove the predicted object. If a coarser category of the predicted object matches an ancestor category of a GT object, RSR assigns a ROLLBACK to replace it with the finest shared ancestor category. Based on these two operations, we prompt an LLM with the original response and the corrected objects to produce a rectified response $y_{\text{rect}} = \text{RSR}(y_{\text{orig}})$. We illustrate an example in Figure 5.

5.2 Preference Construction

We construct two complementary preference sets, $\mathcal{D}_{\text{rel}}$ for reliability preferences and $\mathcal{D}_{\text{gran}}$ for granularity preferences. For each image I, we sample K candidate responses $\mathcal{Y}_{\text{orig}}(I) = \{y_{\text{orig}}^{(k)}\}_{k=1}^{K}$ and obtain K rectified responses $\mathcal{Y}_{\text{rect}}(I) = \{\text{RSR}(y_{\text{orig}}^{(k)})\}_{k=1}^{K}$.

Reliability preferences encourage models to avoid category-incompatible predictions by preferring RSR-rectified responses over their original versions. We construct $\mathcal{D}_{\text{rel}} = \{(I, y_+, y_-)\}_k$ with $(y_+, y_-) = (y_{\text{rect}}^{(k)}, y_{\text{orig}}^{(k)})$, where $y_+^{(k)} \sim \mathcal{Y}_{\text{rect}}(I)$ and $y_-^{(k)} \sim \mathcal{Y}_{\text{orig}}(I)$.

Granularity preferences encourage more informative responses in the RSR-rectified response pool. We construct $\mathcal{D}_{\text{gran}} = \{(I, y_+, y_-)\}_k$, where $y_+, y_- \sim \mathcal{Y}_{\text{rect}}(I)$ and $\text{F1}_{\text{gran}}(I, y_+) > \text{F1}_{\text{gran}}(I, y_-) + \tau$, with τ filtering out pairs with similar scores.

5.3 Metric-Margin DPO

We optimize the model with a metric-margin DPO objective, which is computed as

$$\mathcal{L} = -\mathbb{E}_{(I,y_+,y_-)\sim\mathcal{D}} \log \sigma\left(\Delta_\theta(I, y_+, y_-) - m\right), \quad (11)$$

where $\Delta_\theta(I, y_+, y_-)$ denotes a reward gap, and m denotes the margin. We assign an instance-specific margin to each preference pair (y_w, y_l). A larger performance gap $\delta \in [0, 1]$ of preference pairs leads to a larger margin, which is modeled as $m = \gamma + \alpha\delta$, where γ and α are hyper-parameters.

The reward gap is defined as $\Delta_\theta(I, y_+, y_-) = r_\theta(I, y_+) - r_\theta(I, y_-)$. $r_\theta(\cdot)$ denotes implicit DPO reward, which is computed as

$$r_\theta(I, y) = \beta \log \frac{\pi_\theta(y \mid I)}{\pi_{\text{ref}}(y \mid I)}, \quad (12)$$

where π_θ is the policy model, π_{ref} is the frozen reference model, and β is the temperature.

The computation of m depends on the preference set. For reliability preferences $(I, y_+, y_-) \in \mathcal{D}_{\text{rel}}$, $m_{\text{rel}} = \gamma_{\text{rel}} + \alpha_{\text{rel}}\delta_{\text{rel}}$. With P in Eq. (6), δ_{rel} is

$$\delta_{\text{rel}} = \frac{\text{P}(y_+) - \text{P}(y_-)}{\max_{(I,y_+,y_-)\in\mathcal{D}_{\text{rel}}} \left(\text{P}(y_+) - \text{P}(y_-)\right)}, \quad (13)$$

where the denominator normalizes the gap into $[0, 1]$. For granularity preferences $(I, y_+, y_-) \in \mathcal{D}_{\text{gran}}$, $m_{\text{gran}} = \gamma_{\text{gran}} + \alpha_{\text{gran}}\delta_{\text{gran}}$. δ_{gran} is

$$\delta_{\text{gran}} = \frac{\text{F1}_{\text{gran}}(y_+) - \text{F1}_{\text{gran}}(y_-)}{\max_{(I,y_+,y_-)\in\mathcal{D}_{\text{gran}}} \left(\text{F1}_{\text{gran}}(y_+) - \text{F1}_{\text{gran}}(y_-)\right)}, \quad (14)$$

where F1_{gran} is defined in Eq. (8). We set γ_{rel} and α_{rel} larger than γ_{gran} and α_{gran} for $\mathcal{D}_{\text{gran}}$, so reliability preferences impose overall larger margins than granularity preferences. This encourages reliability-first optimization while favoring finer-grained responses when reliability is satisfied.

The loss of $\mathcal{D}_{\text{rel}}$ and $\mathcal{D}_{\text{gran}}$ is computed as

$$\begin{aligned} \mathcal{L}_{\text{rel}} &= -\mathbb{E}_{(I,y_+,y_-)\sim\mathcal{D}_{\text{rel}}} \log \sigma\left(\Delta_\theta - m_{\text{rel}}\right), \\ \mathcal{L}_{\text{gran}} &= -\mathbb{E}_{(I,y_+,y_-)\sim\mathcal{D}_{\text{gran}}} \log \sigma\left(\Delta_\theta - m_{\text{gran}}\right). \end{aligned} \quad (15)$$

The final objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rel}} + \lambda\mathcal{L}_{\text{gran}}$, where λ is a hyper-parameter.

6 Experiments

We conduct experiments to answer three questions: (1) How do existing MLLMs perform on reliability-prioritized fine-grained generation under different prompting styles? (2) Can our method improve both reliability and supported granularity across prompting styles? (3) Which components of our method contribute to the improvement?

6.1 Experimental Setup

Benchmark and metrics. We evaluate all models on GRANFACT using the granularity-neutral and granularity-aware metrics defined in Section 4.3. The former measures reliability at any valid hierarchy level, while the latter further accounts for the specificity of reliable descriptions.

Models and prompts. We evaluate a diverse set of open-source and closed-source MLLMs, including InstructBLIP-Vicuna-7B (Dai et al., 2023), InternVL3.5-8B (Zhu et al., 2025), Kimi-K2.6 (Moonshot AI, 2026), Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-8B-Instruct (Bai et al., 2025a), GLM-4.6V (Hong et al., 2025), GPT-5.4 (OpenAI, 2026), and Gemini-3.1-Flash-Lite (Google DeepMind, 2026). For each model, we evaluate three prompt styles: *conservative*, *neutral*, and *aggressive*, which induce different levels of generation specificity. Implementation details, model versions, decoding settings, and full prompts are provided in Appendix D.

Training data. For training, we construct a small auxiliary training set using category labels from the GRANFACT annotations to retrieve and synthesize additional single-object images. This results in 525 structured training images, all of which are disjoint from the GRANFACT evaluation set. More details are provided in Appendix D.4.

6.2 Main Results on GRANFACT

Table 1 summarizes the main results on GRANFACT. We highlight four key observations below.

Reliability–granularity trade-off is prevalent. Aggressive prompting generally increases granularity but reduces reliability across models. For example, compared with the conservative prompt, the aggressive prompt increases $\mathrm{G}_{\mathrm{avg}}$ for Qwen3-VL-8B-Instruct from 0.5358 to 0.6040, while decreasing IR from 0.4664 to 0.3718 and F1 from 0.6838 to 0.6529. This shows that prompting models to be more specific does not necessarily make their fine-grained descriptions reliable.

Image-level reliability remains difficult. Across models, IR is much lower than precision, indicating that image-level reliability remains challenging in open-ended visual description. For example, Gemini-3.1-Flash-Lite reaches a precision of 0.7722 under the conservative prompt, but its IR is only 0.5112. This suggests that a full response can still contain unreliable claims even when much of its generated content is grounded.

Frontier models are stronger in reliability yet fragile in fine-grained generation. Frontier and large-scale reference models generally achieve stronger reliability than comparable-scale open-weight models, with Gemini-3.1-Flash-Lite obtaining the best IR, P, and F1 under the conservative prompt. However, this reliability advantage weakens when the model is pushed toward finer-grained generation: for Gemini-3.1-Flash-Lite, $\mathrm{G}_{\mathrm{avg}}$ increases from 0.5524 to 0.6118 under the aggressive prompt, while IR drops from 0.5112 to 0.4672. This shows that even stronger MLLMs remain vulnerable when generating fine-grained descriptions.

Our method improves fine-grained generation without sacrificing reliability. Compared with Qwen3-VL-8B-Instruct, the proposed RP-DPO method improves both reliability and granularity under the same prompt styles. Under the aggressive prompt, it improves IR from 0.3718 to 0.4017 and $\mathrm{F1}_{\mathrm{gran}}$ from 0.3944 to 0.4107, while also increasing $\mathrm{G}_{\mathrm{avg}}$ from 0.6040 to 0.6225. This indicates that our method can improve fine-grained generation without making the model either overly conservative or prone to unreliable fine-grained claims. Moreover, our method achieves comparable performance to the largest-scale model on the G_{avg} metric, further demonstrating the superiority of our method.

6.3 Ablation Study

We conduct ablations under the aggressive prompt setting and compare the full method with variants that remove RSR, reliability preferences, granularity preferences, or metric-margin optimization.

As shown in Table 2, the reliability-prioritized components are crucial for preserving factual reliability. Specifically, removing RSR, reliability preferences, or metric-margin lowers F1 and GIR, although these variants sometimes yield higher $\mathrm{G}_{\mathrm{avg}}$, indicating more aggressive but less reliable fine-grained descriptions. In contrast, removing granularity preferences improves IR and F1 but sub-

Model	Prompt Style	Granularity-neutral Metrics				Granularity-aware Metrics				
		IR	P	R	F1	GIR	P_{gran}	R_{gran}	$F1_{gran}$	G_{avg}
Comparable-Scale Open-Weight Models										
InstructBLIP-Vicuna-7B	Aggressive	0.1962	0.3897	0.3079	0.3440	0.0904	0.1627	0.1286	0.1437	0.4176
	Neutral	0.3959	0.6125	0.2397	0.3446	0.1352	0.2192	0.0858	0.1233	0.3578
	Conservative	0.3890	0.6211	0.2330	0.3388	0.1323	0.2281	0.0855	0.1244	0.3672
InternVL3.5-8B	Aggressive	0.3133	0.4817	0.5595	0.5177	0.1632	0.2421	0.2812	0.2602	0.5026
	Neutral	0.3873	0.6246	0.5257	0.5709	0.1919	0.3035	0.2554	0.2774	0.4858
	Conservative	0.4062	0.6330	0.5574	0.5928	0.1816	0.2894	0.2548	0.2710	0.4571
Qwen2.5-VL-7B	Aggressive	0.3838	0.6258	0.5551	0.5883	0.2126	0.3416	0.3029	0.3211	0.5458
	Neutral	0.4923	0.6699	0.5130	0.5811	0.2624	0.3529	0.2702	0.3061	0.5268
	Conservative	0.4647	0.6701	0.5130	0.5811	0.2229	0.3229	0.2472	0.2800	0.4819
Qwen3-VL-8B-Instruct	Aggressive	0.3718	0.6591	0.6469	0.6529	0.2450	0.3981	0.3907	0.3944	0.6040
	Neutral	0.4251	0.6865	0.6130	0.6477	0.2635	0.4047	0.3613	0.3818	0.5895
	Conservative	0.4664	0.7198	0.6511	0.6838	0.2573	0.3857	0.3488	0.3663	0.5358
Ours (Qwen3-VL-8B)	Aggressive	0.4017	0.6719	0.6480	0.6597	0.2655	0.4183	**0.4034**	**0.4107**	**0.6225**
	Neutral	0.4897	0.7027	0.6163	0.6567	**0.2938**	**0.4217**	0.3698	0.3941	0.6001
	Conservative	**0.4940**	**0.7242**	**0.6514**	**0.6859**	0.2742	0.3935	0.3539	0.3726	0.5433
Frontier / Large-Scale Reference Models										
Kimi-K2.6	Aggressive	0.4527	0.6682	0.5337	0.5934	0.2202	0.4166	0.3328	0.3700	**0.6235**
	Neutral	0.4647	0.6844	0.4954	0.5747	0.2332	0.4193	0.3035	0.3521	0.6127
	Conservative	0.4923	0.7126	0.5357	0.6116	0.2105	0.3786	0.2846	0.3249	0.5313
GLM-4.6V	Aggressive	0.4057	0.6751	0.7075	0.6910	0.2533	0.3881	0.4067	0.3972	0.5748
	Neutral	0.4603	0.7093	0.6671	0.6876	0.2717	0.3881	0.3649	0.3762	0.5471
	Conservative	0.4922	0.7530	0.7128	0.7323	0.2654	0.3803	0.3600	0.3699	0.5051
GPT-5.4	Aggressive	0.2754	0.5894	0.7008	0.6403	0.1684	0.3478	0.4136	0.3779	0.5901
	Neutral	0.4983	0.7467	0.6318	0.6845	0.2752	0.3936	0.3331	0.3608	0.5271
	Conservative	0.4879	0.7612	0.6777	0.7170	0.2369	0.3616	0.3219	0.3406	0.4750
Gemini-3.1-Flash-Lite	Aggressive	0.4672	0.7510	0.7292	0.7400	0.3096	0.4595	**0.4461**	**0.4527**	0.6118
	Neutral	0.4836	0.7571	0.7019	0.7284	**0.3141**	**0.4622**	0.4285	0.4447	0.6105
	Conservative	**0.5112**	**0.7722**	**0.7424**	**0.7570**	0.3019	0.4265	0.4101	0.4182	0.5524

Table 1: Performance on GRANFACT under different prompt styles. Models are grouped into comparable-scale open-weight models and frontier/large-scale reference models.

Method	IR	F1	GIR	$F1_{gran}$	G_{avg}
Baseline	0.3718	0.6529	0.2450	0.3944	0.6040
Full	0.4017↑	0.6597↑	**0.2655**↑	**0.4107**↑	0.6225↑
w/o RSR	0.3374↓	0.6389↓	0.2317↓	0.3963↑	0.6203↑
w/o Rel.	0.2845↓	0.6102↓	0.1938↓	0.3923↓	0.6429↑
w/o Gran.	0.4223↑	0.6691↑	0.2602↑	0.3870↓	0.5784↓
w/o Margin	0.3754↑	0.6337↓	0.2441↓	0.3881↓	0.6125↑

Table 2: Ablation study under the aggressive prompt setting. Arrows compare each result with the baseline. Bold highlights the best results on GIR and $F1_{gran}$, which jointly account for reliability and granularity.

stantially lowers $F1_{gran}$ and G_{avg}, suggesting that the model becomes overly conservative. Overall, the full method achieves the best GIR and $F1_{gran}$, showing that reliability-oriented components suppress unsupported specificity while granularity-oriented preferences prevent the model from collapsing to coarse descriptions.

7 Conclusion

In this paper, we study reliability-prioritized fine-grained generation in MLLMs, where models should generate the finest visual descriptions that remain reliably supported by the image. We introduce GRANFACT, a granularity-aware benchmark with expert-verified multi-object images with hierarchical coarse-to-fine annotations, together with a hierarchy-aware evaluation algorithm that measures both correctness and supported granularity. The proposed RP-DPO method prioritizes reliable predictions while encouraging finer-grained descriptions when reliability is ensured. Experiments on GRANFACT demonstrate that existing MLLMs still struggle to produce descriptions that are both reliable and fine-grained. Experimental results also demonstrate that the proposed RP-DPO method can improve fine-grained generation while preserving reliability.

Limitations

A limitation of our work is that the evaluation set is relatively small and covers a limited visual domains. Future work will expand the scale and diversity of GRANFACT.

References

Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. 1988. Network flows.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-VL technical report. *Preprint*, arXiv:2502.13923.

Yishuo Cai, Renjie Gu, Jiaxu Li, Xuancheng Huang, Junzhe Chen, Xiaotao Gu, and Minlie Huang. 2025. Mhalo: Evaluating mllms as fine-grained hallucination detectors. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9197–9222.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Wei Chen, Xin Yan, Bin Wen, Fan Yang, Tingting Gao, Di Zhang, and Long Chen. 2025. Decoupling contrastive decoding: Robust hallucination mitigation in multimodal large language models. *Advances in Neural Information Processing Systems*, 38:166734–166756.

Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. 2025. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14077–14094.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. 2024. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2025a. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38.

Yuhan Fu, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Xirong Li. 2025b. Mitigating hallucination in multimodal large language model via hallucination-targeted direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16563–16577.

Google DeepMind. 2026. Gemini 3.1 Flash-Lite model card. Google DeepMind Model Card. Accessed: 2026-05-26.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.

Hulingxiao He, Zijun Geng, and Yuxin Peng. 2026. Fine-r1: Make multi-modal llms excel in fine-grained visual recognition by chain-of-thought reasoning. In *International Conference on Learning Representations*.

Hulingxiao He, Geng Li, Zijun Geng, Jinglin Xu, and Yuxin Peng. 2025. Analyzing and boosting the power of fine-grained visual recognition for multi-modal large language models. In *International Conference on Learning Representations*.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, and 1 others. 2025. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Jeonghwan Kim and Heng Ji. 2024. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6187–6207.

Morton Klein. 1967. A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Zhihang Liu, Chen-Wei Xie, Bin Wen, Feiwu Yu, Pandeng Li, Boqiang Zhang, Nianzu Yang, Zuan Gao, Yun Zheng, and Hongtao Xie. 2025. Capability: A comprehensive visual caption benchmark for evaluating both correctness and thoroughness. *Advances in Neural Information Processing Systems*, 38.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pages 23439–23554.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Moonshot AI. 2026. Kimi K2.6: Advancing open-source coding. Kimi Research Blog. Accessed: 2026-05-26.

OpenAI. 2026. Introducing GPT-5.4. OpenAI Product Release. Accessed: 2026-05-26.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. 2024. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018.

Jie Sun, Junkang Wu, Jiancan Wu, Zhibo Zhu, Xingyu Lu, Jun Zhou, Lintao Ma, and Xiang Wang. 2025. Robust preference optimization via dynamic target margins. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5399–5416.

Bingkui Tong, Jiaer Xia, and Kaiyang Zhou. 2025. Mitigating hallucination in multimodal llms with layer contrastive decoding. *arXiv preprint arXiv:2509.25177*.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Fei Wang, Wenxuan Zhou, James Y Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. mdpo: Conditional preference optimization for multimodal large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8078–8088.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and 1 others. 2023. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*.

Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2025. AlphaDPO: Adaptive reward margin for direct preference optimization. In *Proceedings of the 42nd International Conference on Machine Learning*.

Tz-Ying Wu, Chih-Hui Ho, and Nuno Vasconcelos. 2024. Protect: Prompt tuning for taxonomic open set classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16531–16540.

Wei Wu, Xiaomeng Fan, Yuwei Wu, Zhi Gao, Pengxiang Li, Yunde Jia, and Mehrtash Harandi. 2026. Modality alignment across trees on heterogeneous hyperbolic manifolds. In *International Conference on Learning Representations (ICLR)*.

Qinghao Ye, Xianhan Zeng, Fu Li, Chunyuan Li, and Haoqi Fan. 2025. Painting with words: Elevating detailed image captioning with benchmark and alignment learning. In *International Conference on Learning Representations*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2024. Woodpecker: Hallucination correction for multimodal large language models. *Science China Information Sciences*, 67(12):220105.

Zhihan Yin, Jianxin Liang, Yueqian Wang, Yifeng Yao, Huishuai Zhang, and Dongyan Zhao. 2026. Freak: A fine-grained hallucination evaluation benchmark for advanced mllms. *arXiv preprint arXiv:2603.19765*.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2024. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations*, volume 2024, pages 57153–57180.

Hong-Tao Yu, Yuxin Peng, Serge Belongie, and Xiu-Shen Wei. 2025. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. *arXiv preprint arXiv:2504.14988*.

Hong-Tao Yu, Xiu-Shen Wei, Yuxin Peng, and Serge Belongie. 2026. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. In *International Conference on Learning Representations*.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and 1 others. 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*, pages 13807–13816.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, and 1 others. 2025. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pages 89655–89701.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024a. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations*, volume 2024, pages 56969–56998.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024b. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, and 1 others. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pages 18378–18394.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Theoretical Analysis of Semantic Granularity

This section provides the formal definitions and proof for Theorem 1. The analysis is task-level and model-agnostic. It characterizes the Bayes-optimal semantic risk induced by different semantic granularities, rather than assuming that every particular MLLM is strictly hierarchy-consistent.

A.1 Semantic Hierarchy and Cut Sets

Let $\mathscr{S} = (\mathcal{U}, \mathcal{E})$ be a rooted semantic tree. Each node $u \in \mathcal{U}$ represents a semantic category. We write $u \preceq_{\mathscr{S}} v$ if $u = v$ or u is a descendant of v, or equivalently if u is a refinement of v. Let Λ denote the set of leaf nodes. For each node u, let $\Lambda(u) \subseteq \Lambda$ be the set of leaves under u. Thus,

$$u \preceq_{\mathscr{S}} v \quad \Rightarrow \quad \Lambda(u) \subseteq \Lambda(v). \tag{16}$$

A semantic granularity is represented by a cut set $\Omega \subseteq \mathcal{U}$. A valid cut set satisfies

$$\begin{aligned} &\bigcup_{u \in \Omega} \Lambda(u) = \Lambda, \\ &\Lambda(u) \cap \Lambda(v) = \emptyset, \quad \forall u \neq v \in \Omega. \end{aligned} \tag{17}$$

That is, nodes in Ω form a partition of the leaf-level semantic space.

For two cut sets Ω_f and Ω_c, we say that Ω_f is finer than Ω_c, denoted by $\Omega_f \sqsubseteq \Omega_c$, if for every $u \in \Omega_f$, there exists $v \in \Omega_c$ such that $u \preceq_{\mathscr{S}} v$. Because Ω_c is a cut set, this node v is unique. Otherwise, two distinct nodes in Ω_c would have overlapping leaf sets, contradicting Eq. (17). We therefore define the coarsening map

$$\kappa_{f \to c}(u) = v, \quad u \in \Omega_f,\ v \in \Omega_c,\ u \preceq_{\mathscr{S}} v. \tag{18}$$

A.2 Semantic Correctness and Bayes Risk

We consider a task distribution over image–semantic pairs $(I, \xi^\star)$, where $I \in \mathsf{Img}$ is an image and $\xi^\star \in \Lambda$ is the finest true semantic state. A prediction $u \in \Omega$ is semantically correct if its semantic scope covers the true leaf:

$$\xi^\star \preceq_{\mathscr{S}} u. \tag{19}$$

Equivalently, $\xi^\star \in \Lambda(u)$.

For a cut set Ω, a predictor is a function

$$h_\Omega : \mathsf{Img} \to \Omega. \tag{20}$$

Its semantic risk is defined as

$$\varepsilon(h_\Omega) = \Pr[\xi^\star \not\preceq_{\mathscr{S}} h_\Omega(I)], \tag{21}$$

where the probability is taken over the task distribution. The Bayes-optimal semantic risk under cut Ω is

$$\varepsilon^\star(\Omega) = \inf_{h_\Omega : \mathsf{Img} \to \Omega} \varepsilon(h_\Omega). \tag{22}$$

A.3 Proof of Monotonicity

We restate the theorem below.

Theorem. Let Ω_f and Ω_c be two cut sets of $\mathscr{S}$, where $\Omega_f \sqsubseteq \Omega_c$. Then

$$\varepsilon^\star(\Omega_c) \leq \varepsilon^\star(\Omega_f). \tag{23}$$

Proof. Take any predictor $h_f : \mathsf{Img} \to \Omega_f$. Using the coarsening map in Eq. (18), we construct a coarse-grained predictor

$$h_c(I) = \kappa_{f \to c}(h_f(I)). \tag{24}$$

By definition of $\kappa_{f \to c}$,

$$h_f(I) \preceq_{\mathscr{S}} h_c(I). \tag{25}$$

Therefore, for every image–semantic pair $(I, \xi^\star)$,

$$\xi^\star \preceq_{\mathscr{S}} h_f(I) \quad \Rightarrow \quad \xi^\star \preceq_{\mathscr{S}} h_c(I). \tag{26}$$

In words, whenever the fine-grained prediction is correct, its coarsened prediction is also correct. Equivalently, the error event of h_c is contained in the error event of h_f:

$$\{\xi^\star \not\preceq_{\mathscr{S}} h_c(I)\} \subseteq \{\xi^\star \not\preceq_{\mathscr{S}} h_f(I)\}. \tag{27}$$

Taking probabilities gives

$$\varepsilon(h_c) \leq \varepsilon(h_f). \tag{28}$$

Since h_c is a valid predictor on Ω_c, we have

$$\varepsilon^\star(\Omega_c) \leq \varepsilon(h_c) \leq \varepsilon(h_f). \tag{29}$$

This holds for any predictor $h_f : \mathsf{Img} \to \Omega_f$. Taking the infimum over all such h_f yields

$$\varepsilon^\star(\Omega_c) \leq \varepsilon^\star(\Omega_f). \tag{30}$$

□

A.4 Implications

The theorem shows that moving from a coarse cut to a finer cut cannot decrease the Bayes-optimal semantic risk. Intuitively, a fine-grained prediction can always be mapped to its coarser ancestor without turning a correct prediction into an incorrect one, while the reverse is not guaranteed. Thus, finer-grained generation requires solving a harder semantic decision problem.

This result provides a formal motivation for reliability-prioritized fine-grained generation. Fine-grained descriptions may be more informative, but they should be rewarded only when the additional

specificity remains reliable. The empirical analysis in Section 3.2 complements this theorem by showing that, in real MLLMs, eliciting more specific descriptions is often accompanied by reduced image-level reliability.

The analysis above focuses on a single category claim. Open-ended visual descriptions introduce additional challenges, including multiple objects, quantities, attributes, repeated mentions, omissions, and hallucinated entities.

B Details on Benchmark Construction

B.1 Dataset Statistics

This subsection provides additional statistics of the GRANFACT evaluation set, supplementing the summary visualizations in Figure 4. The benchmark contains 581 images from seven visual domains. Table 3 reports the exact number of images in each domain.

Domain	# Images	Percentage
Daily objects	183	31.5%
Animals	100	17.2%
Cars	80	13.8%
Electronics	75	12.9%
Landmarks	75	12.9%
Games	43	7.4%
Plants	25	4.3%
Total	581	100.0%

Table 3: Domain distribution of the GRANFACT evaluation set.

We further summarize the entity-level annotation statistics in Table 4. Here, an annotated entity corresponds to a GT entry g_j in the structured annotation of an image. These statistics describe the number of annotated entities per image and provide a more detailed view of the multi-entity structure of the evaluation set.

Statistic	Value
Total annotated GT entities	2302
Average GT entities per image	3.96
Median GT entities per image	4
Minimum GT entities per image	1
Maximum GT entities per image	14
Images with 1 annotated entity	124
Images with 2 annotated entities	116
Images with 3 or more annotated entities	341

Table 4: Entity-level statistics of GRANFACT.

Table 5 reports the distribution of category-path depths. For each annotated entity g_j, the depth is defined as the length L_j of its coarse-to-fine category path $\mathcal{H}_j = (c_{j,1}, \ldots, c_{j,L_j})$. A larger depth indicates that the entity is annotated with a more detailed semantic hierarchy.

Category-path depth L_j	# Entities	Percentage
2	7	0.3%
3	175	7.6%
4	522	22.7%
5	880	38.2%
6	581	25.2%
7	123	5.3%
8	12	0.5%
9	2	0.1%
Total	2302	100.0%

Table 5: Distribution of category-path depths over annotated GT entities.

In addition to category labels, GRANFACT includes non-quantitative visual attributes for annotated entities when such attributes are visually identifiable and useful for evaluation. Table 6 summarizes the attribute annotations.

Statistic	Value
Total attribute annotations	13144
Average attributes per annotated entity	5.71

Table 6: Attribute annotation statistics of GRANFACT.

B.2 Image Collection, Filtering, and Annotation Protocol

Since the goal of GRANFACT is to evaluate reliability-prioritized fine-grained generation, we collected and annotated images with an emphasis on fine-grained visual understanding. We recruited 10 domain-aware annotators to participate in the collection and annotation process. The annotators were familiar with at least one of the covered visual domains, such as daily objects, animals, plants, cars, electronics, landmarks, or games. The process was conducted iteratively: candidate images were first collected according to the benchmark goal, then preliminarily annotated with entity-level category paths and visual attributes, and finally reviewed and filtered before inclusion in the final evaluation set.

During collection, annotators followed several practical guidelines to ensure that the images were suitable for evaluating fine-grained visual description. First, each retained image should contain at least one visually identifiable entity that can be

associated with a coarse-to-fine category path. Second, when available, annotators preferred images with diverse entity compositions, visually related entities, or realistic visual contexts, since such cases provide useful open-ended evaluation settings. Images centered on a single entity were also retained when the entity could be annotated at a sufficiently fine-grained level with reasonable confidence. Third, for each annotated entity, its finest category label should be reasonably supported by the image and, when needed, by common visual knowledge or reference information available to the annotators. Images were excluded when the intended fine-grained identities of key entities were too uncertain to annotate reliably.

The initial collection contained approximately 700 candidate image–annotation pairs. Each candidate pair consisted of an image and preliminary structured annotations. For each annotated entity, the preliminary annotation included a coarse-to-fine category path and a set of visual attributes. The category path records the semantic granularity of the entity from a coarse category to the finest category that could be reasonably identified. The attributes describe visible non-quantitative properties such as color, material, posture, spatial position, shape, or other domain-specific visual cues when applicable.

We manually reviewed the candidate samples and revised or removed samples with clear annotation issues. The review focused on whether the annotated entities were visually present, whether their category paths were appropriate, whether the finest category labels were sufficiently supported, and whether the visual attributes were consistent with the image.

We filtered out candidate samples according to the following criteria:

- **Low visual quality:** images that were blurry, low-resolution, over-compressed, or otherwise difficult to inspect.
- **Insufficient visual evidence:** images where the intended fine-grained category of a key entity could not be determined with reasonable confidence.
- **Ambiguous labels:** images where multiple fine-grained labels appeared plausible for a key entity or where a stable category path was difficult to define.
- **Severe occlusion or truncation:** images where key entities were heavily occluded, cropped, or too small to support reliable annotation.
- **Duplicate or near-duplicate samples:** visually redundant images that contributed little additional diversity.
- **Unsuitable scenes:** images that did not contain identifiable fine-grained entities or were not well aligned with the intended open-ended visual description setting.

When inconsistencies were identified, the annotations were revised when a reliable correction was clear; otherwise, the sample was removed from the benchmark. After this filtering and review process, we retained 581 image–annotation pairs as the final GRANFACT evaluation set. This process aims to improve annotation reliability while preserving the fine-grained and visually diverse characteristics needed for evaluating open-ended model responses.

B.3 Annotator Instructions, Consent, and Ethical Considerations

This subsection provides additional details about the human annotation process used for constructing GRANFACT. The annotation task was limited to image collection, image filtering, and object-level visual annotation for research purposes. It did not involve interventions with human subjects or the collection of sensitive personal information from annotators.

Annotator recruitment. We recruited 10 domain-aware annotators to participate in image collection, filtering, and annotation. Annotators were selected based on their familiarity with at least one of the covered visual domains, including daily objects, animals, plants, cars, electronics, landmarks, and games. We did not use an external crowdsourcing platform. The annotators were instructed to work on domains for which they had sufficient familiarity, and uncertain cases were reviewed or filtered out during the quality-control stage.

Annotation instructions. Annotators were given the following instructions.

> You will help construct a research benchmark for evaluating fine-grained visual description in multimodal large language

models. For each candidate image, first determine whether the image is suitable for fine-grained visual description evaluation. A suitable image should contain at least one visually identifiable entity that can be associated with a coarse-to-fine category path. Prefer images with diverse object compositions, visually related entities, or realistic visual contexts. Images centered on a single object may also be retained if the object can be annotated at a sufficiently fine-grained level with reasonable confidence.

For each retained image, annotate the visible target entities. For each entity, provide: (1) a coarse-to-fine category path, from a broad category to the finest category that can be reasonably supported; (2) the quantity of visible instances; and (3) visible non-quantitative attributes, such as color, material, shape, pose, spatial position, or other domain-specific visual cues when applicable.

Do not guess fine-grained identities that are not supported by the image. If the finest category of an entity cannot be determined with reasonable confidence, annotate the entity only up to the finest reliable level, or mark the sample for review. When needed, use common visual knowledge or reference information to verify fine-grained categories, but do not add labels that remain visually ambiguous. Coarse but reliable annotations are preferred over unsupported fine-grained labels.

Exclude candidate images if they are blurry, low-resolution, over-compressed, severely occluded, heavily cropped, or otherwise difficult to inspect. Also exclude images where the intended fine-grained category is too uncertain, where multiple fine-grained labels are plausible, where a stable category path cannot be defined, or where the scene is not suitable for open-ended visual description evaluation. Duplicate or near-duplicate images should be removed.

The collected annotations will be used for constructing a research benchmark and for reporting aggregate statistics and examples in the paper. The task focuses on object-level visual annotation and should not require annotators to provide demographic, sensitive, or private personal information.

Review and quality control. Candidate image–annotation pairs were manually reviewed before inclusion in the final evaluation set. The review focused on whether annotated entities were visually present, whether their category paths were appropriate, whether the finest category labels were sufficiently supported, and whether the visual attributes were consistent with the image. Samples with unclear visual evidence, ambiguous fine-grained labels, severe occlusion or truncation, low visual quality, or duplicate content were revised or removed.

Consent and data use. Annotators were informed that the annotations would be used to construct a research benchmark, evaluate multimodal models, and report aggregate statistics and qualitative examples in the paper. The benchmark focuses on objects and scenes rather than human subjects. During image selection, we avoided images centered on identifiable people or sensitive personal information. For real-world photographs, images were collected by the research team or used with permission. For web-sourced images, redistribution will follow the corresponding source licenses or terms of use. When redistribution rights are unclear, we will release annotations, metadata, and evaluation code as appropriate rather than redistributing the original images.

Ethics review. The study did not undergo formal ethics board review. The annotation task involved low-risk object-level image annotation, did not involve interventions with human subjects, and did not require the collection of sensitive personal information. We nevertheless provided annotators with task instructions, informed them of the research use of the annotations, and filtered images to avoid sensitive or privacy-invasive content.

Annotator compensation and crediting. The annotation was conducted as part of the research project by domain-aware annotators rather than through an external crowdsourcing marketplace. Annotators were compensated or credited according to the project arrangement. No annotator was

asked to provide sensitive personal information as part of the annotation task.

B.4 More Qualitative Examples

We provide additional qualitative examples of GRANFACT annotations in Figure 6, Figure 7, and Figure 8. These examples complement the qualitative examples shown in the main paper and cover the remaining domains, including landmarks, cars, daily objects, plants, and games.

C Evaluation Details

C.1 Details of Response Parsing

GRANFACT evaluates open-ended model responses rather than closed-set predictions. Before applying the hierarchy-aware assignment algorithm, we first convert each free-form response into a set of structured prediction entities. Given an MLLM response, we use an LLM parser to extract the primary physical objects mentioned in the response, together with their category descriptions and visual attributes.

A rule-based parser is brittle in this setting because open-ended responses often express object information across multiple clauses or sentences. For example, a model may introduce an object in one sentence and describe its color, position, quantity, or fine-grained identity later in the response. Responses may also contain coordinated noun phrases, implicit quantities, uncertain category claims, or background objects that should not be treated as primary predictions. We therefore use an LLM parser to leverage its language-understanding ability while constraining its output with a fixed extraction prompt.

In our implementation, we use Qwen3.5-27B as the parser model. The parser is instructed to extract only in-scope primary physical objects, keep quantity and position as attributes, avoid extracting generic background or low-information context objects, and output a JSON array following the specified format. The resulting prediction entities are then used as the input to the prediction–GT assignment algorithm described in Section 4.2.2. The core parsing prompt is shown in Figure 9. In implementation, we prepend a short domain-specific scope instruction to restrict extraction to the target visual domain, while keeping the same output format and general extraction rules across domains.

C.2 Granularity-Aware Assignment Score Computation

This subsection provides additional details on how we compute the granularity-aware assignment score w_{ij} in Eq. (4). For each parsed prediction entity p_i and each GT entity g_j, the score is determined by two components: the matched category granularity level ℓ_{ij} and the attribute consistency score a_{ij}. Both components are obtained with an LLM judge.

Category granularity level. Let q_i denote the predicted category of p_i, and let $\mathcal{H}_j = (c_{j,1}, \ldots, c_{j,L_j})$ denote the coarse-to-fine category path of GT entity g_j. The goal is to determine the finest level in $\mathcal{H}_j$ that is supported by the prediction category q_i. We denote this level as $\ell_{ij} \in \{0, \ldots, L_j\}$, where $\ell_{ij} = 0$ indicates that p_i is category-incompatible with g_j, and $\ell_{ij} > 0$ indicates that p_i can be matched to the ℓ_{ij}-th node in the GT category path.

We use an LLM judge to compare q_i with the full GT category path $\mathcal{H}_j$ and determine ℓ_{ij}. The judge is instructed to consider synonyms, paraphrases, and naming variants that refer to the same semantic scope. For example, a prediction such as "cell phone" may match a GT node such as "smartphone" if they describe the same category scope. The judge is also instructed to reject mutually exclusive fine-grained claims. For instance, if a GT entity is annotated as "smartphone" → "iPhone" → "iPhone 15 Pro Max", a prediction such as "iPhone 15 Pro" should not receive credit merely because it shares the coarser ancestor "iPhone". In such cases, the prediction makes an incompatible fine-grained commitment and is treated as category-incompatible with that GT entity.

More generally, ℓ_{ij} reflects the finest GT-supported semantic scope that the prediction can reliably claim. A coarse but correct prediction receives a positive but lower matched level, while a correct fine-grained prediction receives a higher matched level. A prediction that cannot be grounded in any node of the GT category path receives $\ell_{ij} = 0$. The exact category-matching prompt is shown in Figure 10.

Attribute consistency score. For category-compatible pairs with $\ell_{ij} > 0$, we further compute an attribute consistency score $a_{ij} \in [0, 1]$. This score measures how well the non-quantitative attributes asserted by the prediction entity p_i are sup-

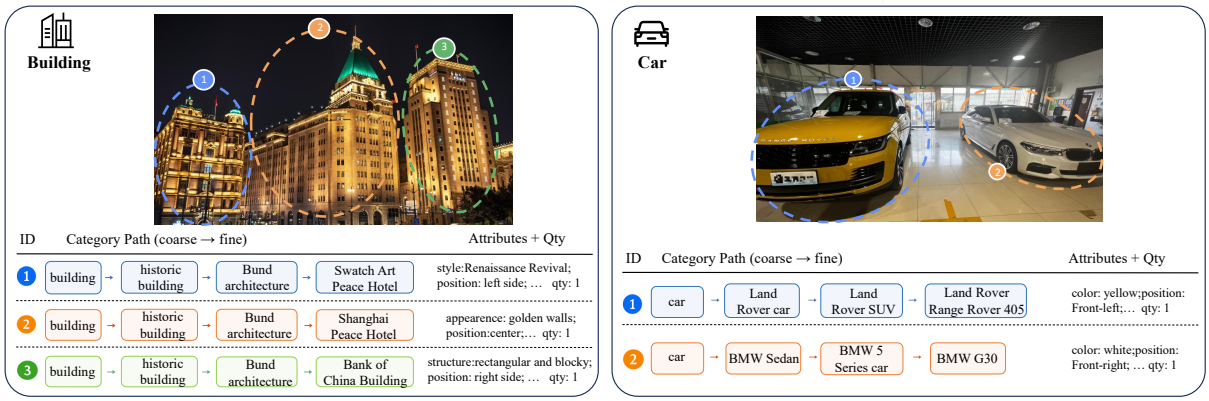

Figure 6: Additional qualitative examples from the landmark and car domains.

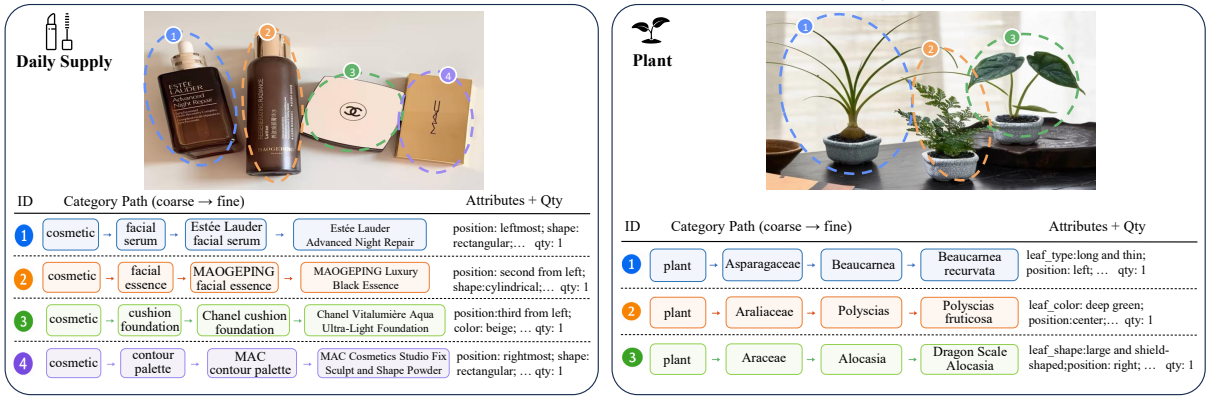

Figure 7: Additional qualitative examples from the daily-object and plant domains.

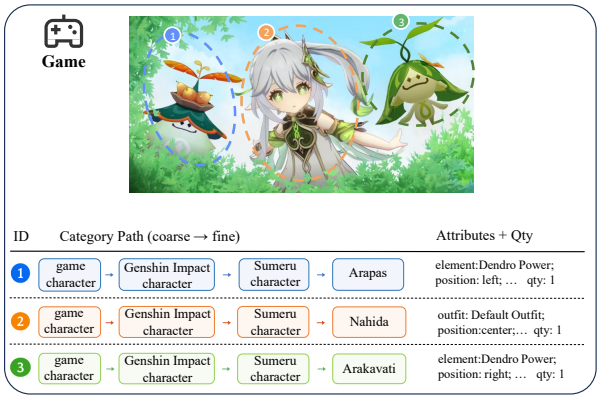

Figure 8: Additional qualitative example from the game domain.

ported by the GT entity g_j. The attributes include properties such as color, material, texture, shape, posture, spatial position, visible state, brand-related visual cues, and other domain-specific visual details when applicable. Quantity is not evaluated as part of a_{ij}; it is handled separately by the assignment constraints through the predicted quantity s_i and GT capacity d_j.

We provide the LLM judge with one predicted object and all category-compatible GT candidates. For each candidate g_j, the judge identifies which GT attribute facts are explicitly covered by the predicted attributes of p_i. Let $\mathcal{A}_i^{\text{pred}}$ denote the set of non-quantitative attributes predicted for p_i, let $\mathcal{A}_j^{\text{gt}}$ denote the set of non-quantitative GT attributes of g_j, and let $\mathcal{A}_{ij}^{\text{cov}} \subseteq \mathcal{A}_j^{\text{gt}}$ denote the GT attributes of g_j that are covered by p_i. We compute

$$\mathrm{P}_{ij}^{\text{attr}} = \frac{|\mathcal{A}_{ij}^{\text{cov}}|}{|\mathcal{A}_i^{\text{pred}}|}, \tag{31}$$

$$\mathrm{R}_{ij}^{\text{attr}} = \frac{|\mathcal{A}_{ij}^{\text{cov}}|}{|\mathcal{A}_j^{\text{gt}}|}, \tag{32}$$

and set

$$a_{ij} = \frac{2\mathrm{P}_{ij}^{\text{attr}}\mathrm{R}_{ij}^{\text{attr}}}{\mathrm{P}_{ij}^{\text{attr}} + \mathrm{R}_{ij}^{\text{attr}}}. \tag{33}$$

When both p_i and g_j have no non-quantitative attributes, we set $a_{ij} = 1$, so that the edge score reflects a pure category-level match. If $\mathrm{P}_{ij}^{\text{attr}} + \mathrm{R}_{ij}^{\text{attr}} = 0$ in other edge cases, we set $a_{ij} = 0$. Our implementation additionally logs an auxiliary contradiction flag for debugging, but it is not used in computing a_{ij}.

For category-incompatible pairs with $\ell_{ij} = 0$, the attribute score does not affect the final assignment score, since the edge is already treated as incompatible at the category level. The LLM judge rubric for attribute scoring is shown in Figure 11.

Assignment score. Given ℓ_{ij} and a_{ij}, we compute the assignment score w_{ij} as in Eq. (4). For a real GT entity $j \leq n$, a category-compatible pair receives

$$w_{ij} = \frac{\ell_{ij} + a_{ij}}{L_j + 1}, \quad \ell_{ij} > 0. \tag{34}$$

This normalization keeps the score in a comparable range across GT entities with different category-path depths. It also ensures that matching a finer category level receives a higher score than matching a coarser level, while attribute consistency can further distinguish assignments at the same matched category level.

If p_i is category-incompatible with a real GT entity g_j, we set $w_{ij} = -1$. This negative score discourages the optimizer from assigning unsupported predictions to real GT entities. For the dummy hallucination entity g_{n+1}, we set $w_{i,n+1} = 0$. Therefore, unsupported prediction quantity can be assigned to the dummy entity instead of being forced to match a real GT entity. This design makes hallucinated or over-specific content explicitly count as unsupported while avoiding spurious matches to real objects.

Efficient implementation. Conceptually, the LLM judge can be applied to every prediction–GT pair (p_i, g_j) to obtain ℓ_{ij} and a_{ij}. In practice, many GT entities in the same image may share coarse category nodes, which can lead to redundant category-level judgments. To reduce such redundancy, we provide an efficient implementation while preserving the same prediction–GT scoring principle described above. In particular, a prediction is considered compatible with a real GT entity only when the judge determines that the match is scope-consistent and does not contain a mutually exclusive fine-grained claim. After the granularity levels and attribute consistency scores are obtained, the resulting score matrix $\mathbf{W} = [w_{ij}]$ is used by the capacity-constrained assignment solver described in Appendix C.3.

C.3 Capacity-Constrained Assignment Solver

This subsection provides implementation details for solving the lexicographic assignment problem in Eq. 3. Given the prediction entities, GT entities, and the granularity-aware assignment scores, we instantiate the prediction–GT alignment as a capacity-constrained bipartite flow problem.

Graph construction. For each parsed prediction entity p_i, we create a prediction node with supply s_i. For each GT entity g_j, we create a GT node with capacity d_j. A directed edge is added from p_i to g_j only when the pair is category-compatible, i.e., when $\ell_{ij} > 0$. The edge capacity is set to

$$u_{ij} = \min(s_i, d_j), \tag{35}$$

so that the assigned quantity on this edge cannot exceed either the prediction supply or the GT capacity. We then add a source node connected to all prediction nodes and a sink node connected from all GT nodes. The source-to-prediction edge has capacity s_i, and the GT-to-sink edge has capacity d_j.

In Section 4.2.2, we introduce a dummy hallucination entity g_{n+1} to present the assignment problem in a unified form: every predicted quantity is either assigned to a real GT entity or to the dummy entity. In implementation, we use an equivalent real-GT-only graph and do not explicitly instantiate the dummy node. Predicted quantity that cannot be assigned to any category-compatible real GT entity remains unmatched. This unmatched residual can be used for diagnostic accounting of unsupported or over-counted predictions, while the matched quantities on real GT edges determine the assignment scores used by the main metrics.

Lexicographic objective. The assignment objective is lexicographic: it first maximizes the total category-compatible matched quantity $M(\mathbf{X})$, and then, among assignments with the same matched quantity, maximizes the granularity-aware score $M_{\text{gran}}(\mathbf{X})$. We implement this priority using a standard large-constant transformation in the min-cost-flow objective.

For each compatible edge (p_i, g_j), we assign a reward consisting of two parts: a large base reward for assigning one unit of predicted quantity to a real GT entity, and a smaller granularity-aware reward based on the edge score w_{ij}. Since the solver minimizes cost, this reward is represented as a negative edge cost:

$$\text{cost}_{ij} = -\left(B + \tilde{w}_{ij}\right), \tag{36}$$

where $\tilde{w}_{ij}$ is an integer-scaled version of w_{ij} and B is a large constant.

The constant B is chosen so that the base reward for one additional matched quantity is larger than any possible difference in total granularity-aware utility among assignments with the same or fewer matched quantities. Therefore, the solver always prefers an assignment with larger category-compatible matched quantity. Only when two assignments match the same total quantity does the second term $\tilde{w}_{ij}$ determine which assignment is preferred. This realizes the lexicographic objective in Eq. 3 within a single min-cost-flow problem.

Successive shortest augmenting path. We solve the resulting min-cost-flow problem with a successive shortest augmenting-path algorithm on the residual network. Starting from zero flow, the algorithm repeatedly finds the shortest source-to-sink path with negative total cost in the residual graph. If such a path exists, it augments as much flow as possible along that path, updates the residual capacities, and continues. The procedure stops when no negative-cost source-to-sink path remains. At this point, no further assignment can increase the lexicographic objective.

The returned flow specifies the assigned quantity x_{ij} for each activated prediction–GT edge. For each GT entity, the incoming flow determines how much of its annotated quantity is covered by model predictions. For each prediction entity, any quantity not assigned to a real GT entity does not contribute to M_z or $M_{\text{gran},z}$; it only remains unmatched while the total predicted quantity is still included in S_z. Thus, unmatched prediction quantity affects the final metrics by reducing the fraction of predicted content that is grounded, rather than by adding a separate score term. The quantities S_z, D_z, M_z, and $M_{\text{gran},z}$ are then used for metric computation.

Validity checks. After solving, we verify that the produced assignment satisfies the capacity constraints. Specifically, the assigned quantity on each edge must not exceed its edge capacity, the total outgoing assigned quantity of a prediction entity must not exceed its predicted supply, and the total incoming assigned quantity of a GT entity must not exceed its GT capacity. These checks ensure that the resulting assignment is a feasible capacity-constrained alignment before metric computation.

C.4 Metric Computation and Well-Definedness

This subsection provides additional details on the computation of the metrics introduced in Section 4.3. We first describe how example-level quantities are aggregated at the dataset level, and then explain why the reported metrics are well-defined even when the optimal assignment matrix in Eq. (3) is not unique.

Dataset-level aggregation. For each evaluation example $z \in \mathcal{Z}$, let S_z denote the total predicted quantity parsed from the model response, and let D_z denote the total annotated GT quantity. After solving the prediction–GT assignment problem, we obtain the granularity-neutral score M_z and the granularity-aware matched score $M_{\text{gran},z}$. Dataset-level precision, recall, and F1 are computed by micro-averaging these quantities over all examples:

$$\begin{aligned} S &= \sum_{z\in\mathcal{Z}} S_z, & D &= \sum_{z\in\mathcal{Z}} D_z, \\ M &= \sum_{z\in\mathcal{Z}} M_z, & M_{\text{gran}} &= \sum_{z\in\mathcal{Z}} M_{\text{gran},z}. \end{aligned} \tag{37}$$

The metrics in Section 4.3 are then computed from these aggregated quantities. This micro-averaged computation treats each predicted or annotated quantity unit consistently across the dataset, rather than first computing per-image scores and then averaging them.

Granularity-neutral metrics. Granularity-neutral metrics evaluate whether the predicted entities can be grounded in category-compatible GT entities, regardless of how fine-grained the grounded descriptions are. In particular, M counts the amount of predicted quantity assigned to category-compatible GT entities. Thus, precision measures the fraction of generated quantity that is grounded, while recall measures the fraction of GT quantity that is covered by grounded predictions:

$$\mathrm{P} = \frac{M}{S}, \quad \mathrm{R} = \frac{M}{D}, \quad \mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P}+\mathrm{R}}. \tag{38}$$

When the denominator of F1 is zero, we set F1 to zero. In addition to these quantity-level metrics, we report the Image Reliability Rate IR, which is a stricter image-level metric. An example contributes positively to IR only when all predicted quantity in that response is assigned to category-compatible GT entities:

$$\mathrm{IR} = \frac{1}{|\mathcal{Z}|} \sum_{z\in\mathcal{Z}} \mathbb{I}[M_z = S_z]. \tag{39}$$

Therefore, P, R, and F1 measure quantity-level grounding, while IR measures whether a complete open-ended response is free from unsupported predicted quantity.

Granularity-aware metrics. Granularity-aware metrics further account for the specificity and attribute consistency of grounded predictions. Instead of counting each category-compatible assignment equally, M_{gran} weights grounded assignments by their matched category level and attribute consistency. A prediction receives a higher score when it is matched to a finer supported category level and has more consistent non-quantitative attributes. Based on M_{gran}, we compute granularity-weighted precision, recall, and F1:

$$\begin{aligned} \mathrm{P}_{\text{gran}} = \frac{M_{\text{gran}}}{S}, \quad \mathrm{R}_{\text{gran}} = \frac{M_{\text{gran}}}{D}, \\ \mathrm{F1}_{\text{gran}} = \frac{2\mathrm{P}_{\text{gran}}\mathrm{R}_{\text{gran}}}{\mathrm{P}_{\text{gran}} + \mathrm{R}_{\text{gran}}}. \end{aligned} \tag{40}$$

Again, when the denominator of $\mathrm{F1}_{\text{gran}}$ is zero, we set $\mathrm{F1}_{\text{gran}}$ to zero.

We also report the granularity-aware counterpart of image reliability:

$$\mathrm{GIR} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \mathbb{I}[M_z = S_z] \cdot \frac{M_{\text{gran},z}}{M_z}, \tag{41}$$

where the granularity term is set to zero when $M_z = 0$. This metric gives non-zero credit only to responses that are fully reliable under the granularity-neutral criterion, and further weights such responses by their average granularity-aware score. Finally, we report

$$\mathrm{G}_{\text{avg}} = \frac{M_{\text{gran}}}{M}, \tag{42}$$

where G_{avg} is set to zero when $M = 0$. This diagnostic metric measures the average granularity-aware score among grounded predictions. It helps distinguish a model that is reliable but mostly produces coarse descriptions from one that is reliable and also more specific.

Well-definedness under non-unique assignments. The lexicographic optimization in Eq. (3) may admit multiple optimal assignment matrices. This non-uniqueness can occur when several GT entities receive exactly the same assignment scores, or when multiple assignments lead to the same reliability and granularity-aware objectives. However, the reported metrics do not depend on which optimal assignment matrix is selected.

To see this, consider an example z and let $\mathcal{X}_z^\star$ denote the set of lexicographically optimal assignment matrices for this example. By definition, every $\mathbf{X}^\star \in \mathcal{X}_z^\star$ first achieves the same maximal value of the primary objective $M(\mathbf{X})$. Therefore, M_z is invariant across all optimal assignments. Among the assignments that maximize $M(\mathbf{X})$, the second stage maximizes $M_{\text{gran}}(\mathbf{X})$. Thus, every lexicographically optimal assignment also achieves the same maximal value of $M_{\text{gran}}(\mathbf{X})$, making $M_{\text{gran},z}$ invariant as well.

The remaining quantities, S_z and D_z, are determined directly by the parsed model response and the GT annotations, and do not depend on the assignment matrix. Consequently, the example-level scalar quantities used by the metrics, i.e., S_z, D_z, M_z, and $M_{\text{gran},z}$, are uniquely determined even if the concrete optimal assignment matrix is not unique. Since all reported metrics are functions only of these scalar quantities, the benchmark scores are well-defined. Non-uniqueness may affect which tied assignment is chosen for qualitative visualization, but it does not affect any reported quantitative metric.

C.5 Uncertain Quantity Handling

Some images contain objects whose exact quantities are difficult to determine. For example, products on a crowded shelf may be heavily occluded, partially visible, or arranged in a way that makes exact counting unreliable. Model responses may also use vague quantity expressions such as "several", "many", or "a group of". To avoid treating all such cases as exact counting errors, we explicitly mark uncertain quantities and handle them separately during assignment and metric computation.

Quantity normalization. For both parsed predictions and GT annotations, each entity is associated with a quantity type. If an entity has an explicit count, we treat it as a numeric quantity. If no count is specified, we use a default quantity of one. If the quantity is expressed as uncertain, approximate, ranged, or plural-but-unspecified, we mark its quantity type as uncertain. The uncertain label means that the entity should not be interpreted as having a fixed exact count, even though a temporary unit value can be used for the initial assignment step.

Initial assignment. The capacity-constrained solver requires finite supplies and capacities. Therefore, in the initial flow problem, uncertain quantities are temporarily instantiated with unit supply or capacity, while their uncertain quantity type is retained. This gives the solver a conservative initial assignment without committing to a precise count. The standard capacity constraints and granularity-aware edge utilities are then applied as described

in Appendix C.3.

Post-assignment repair. After the initial assignment, we apply a limited repair step for uncertain quantities. The repair is augment-only: it does not re-solve the full assignment problem, but only expands uncertain quantities along existing category-compatible edges.

First, if a prediction has remaining unmatched quantity and it is connected to a GT entity whose quantity is uncertain, the uncertain GT capacity may be expanded to absorb the residual prediction quantity. Intuitively, when the GT annotation indicates that the exact number is uncertain, extra compatible predicted instances should not necessarily be treated as over-counting errors.

Second, if a GT entity has remaining uncovered quantity and it is connected to a prediction whose quantity is uncertain, the uncertain prediction supply may be expanded to cover the residual GT quantity. This reflects the fact that a vague prediction such as "several bottles" can cover more than one compatible GT instance when the exact predicted count is not specified.

When multiple compatible edges are available for such repair, we prefer edges with higher granularity-aware utility and stronger attribute agreement, using a deterministic tie-breaking rule. This keeps the repair consistent with the assignment objective while avoiding arbitrary allocation of uncertain quantities.

Credit discount for uncertain predictions. Uncertain quantities on the prediction side are handled conservatively in metric computation. If a prediction has uncertain quantity and is matched to a GT entity with a certain quantity, we apply a partial credit discount to the matched true-positive quantity. If both the prediction and the GT entity have uncertain quantities, no discount is applied.

Formally, for an assigned prediction–GT edge (p_i, g_j) with assigned quantity x_{ij}, we define an effective matched quantity

$$\widehat{x}_{ij} = \rho_{ij} x_{ij}, \tag{43}$$

where we set $\rho_{ij} = \lambda_{\text{unc}}$ only when p_i has uncertain quantity and g_j has certain quantity; otherwise, $\rho_{ij} = 1$. Thus,

$$\rho_{ij} = \begin{cases} \lambda_{\text{unc}}, & \text{unc}(p_i) \wedge \neg\text{unc}(g_j), \\ 1, & \text{otherwise}, \end{cases} \tag{44}$$

where $\text{unc}(\cdot)$ indicates uncertain quantity status. We set $\lambda_{\text{unc}} = 0.5$ in our implementation. Thus, an uncertain prediction receives partial credit when matched to a GT entity with a known exact count, because the prediction is category-compatible but does not make an exact quantity commitment. In contrast, when both sides have uncertain quantities, the match is not penalized for quantity uncertainty.

The effective matched quantity $\widehat{x}_{ij}$ is used when accumulating matched credit for the reliability-oriented metrics. The total predicted quantity S_z still counts the prediction-side quantity, so uncertain predictions do not receive free precision. They only receive reduced matched credit when their quantity uncertainty is less specific than the GT quantity.

Metric accounting. Uncertain quantity handling does not introduce a separate metric. Instead, it affects the quantities used by the existing metrics in Section 4.3. Matched real-GT assignments contribute to the matched quantities according to their effective credit. Prediction quantity that remains unmatched after assignment and repair does not contribute to M_z or $M_{\text{gran},z}$, while the corresponding predicted quantity is still included in S_z. Therefore, unmatched prediction quantity lowers precision by reducing the fraction of predicted content that is grounded, rather than by adding a separate penalty term.

Similarly, uncovered GT quantity affects recall through D_z and the matched quantity. If an uncertain prediction is expanded to cover compatible GT residual quantity, the additional matched quantity is credited according to the discount rule above. This allows vague quantity predictions to receive appropriate credit when they are compatible with the GT, while still penalizing them when the GT provides a more precise count.

Uncertain quantities outside the solver. Some entities do not enter the capacity-constrained solver. For example, a prediction may be rejected before assignment because its category is incompatible with all GT entities, or a GT entity may have no compatible prediction. When such entities have uncertain quantities, we estimate their effective quantities using the average quantity of comparable entities with known counts in the same evaluation instance when available, and fall back to a unit quantity otherwise. This convention prevents uncertain quantities outside the solver from being arbitrarily treated as exact large counts, while still allowing them to contribute to the precision or recall denominator in a controlled way.

D Detailed Experimental Settings

D.1 Evaluated Models

We evaluate both open-source and closed-source MLLMs to cover a broad range of model families, model scales, and deployment settings. Table 7 summarizes the evaluated models and how they are accessed. For locally evaluated models, we load the corresponding HuggingFace checkpoints. For API-based models, we use the API-accessible versions available at the time of evaluation. All models are evaluated with the same set of images, prompt styles, and evaluation pipeline.

For locally evaluated models, we follow the official model-specific conversation template and image preprocessing pipeline when available. For API-based models, we submit the same image and text prompt through the corresponding API interface. We do not apply model-specific prompt engineering beyond adapting the input format required by each interface. The three evaluation prompt styles are kept semantically identical across models and are provided in Appendix D.3. Decoding and inference settings are described in Appendix D.2.

D.2 Decoding and Inference Settings

For each image–prompt pair, we generate one response from each model. All prompt styles use the same decoding configuration within each model, so that differences among conservative, neutral, and aggressive prompts are induced by the prompt instructions rather than by decoding changes.

For locally evaluated models, we use deterministic decoding with sampling disabled, temperature set to 0, and a maximum generation length of 2048 new tokens. Since sampling is disabled, sampling-specific parameters such as top-p do not affect generation. For API-based models, we use the closest available deterministic setting provided by the corresponding API and set the maximum output length to 2048 tokens when the option is available.

For locally evaluated models, we use the official conversation template and default image preprocessing when available. For API-based models, we use the corresponding API interface and only adapt the input format as required by the provider. All generated responses are processed by the same response parsing, assignment, and metric computation pipeline described in Appendix C.

Reporting. Unless otherwise stated, all reported results are aggregate metrics computed over the full GRANFACT evaluation set; model performance is reported from a single evaluation run under the specified prompt style and decoding settings. We do not report error bars, standard deviations, or confidence intervals across multiple random seeds or repeated runs. To reduce stochastic variation, we use deterministic decoding whenever available, including disabling sampling and setting temperature to 0 for locally evaluated models.

D.3 Evaluation Prompts

We evaluate each model under three prompt styles to study how different levels of requested specificity affect reliability and granularity. The neutral prompt asks for a standard image description, the conservative prompt encourages factual description without guessing, and the aggressive prompt encourages more detailed generation. All prompts require English responses and are applied consistently across models.

D.4 Auxiliary Training Data Construction

To train our alignment method, we construct a small auxiliary training set that is separate from the GRANFACT evaluation set. The goal is to provide fine-grained, structured training examples without using the evaluation images. We focus on five domains: daily objects, animals, plants, electronics, and cars.

We first collect textual category labels from these domains and use them as image search queries. For each category label, we retrieve five candidate images from the web. The candidate images are first screened by an MLLM, Qwen3.5-27B, to remove clearly mismatched or visually unsuitable images. They are then reviewed by human annotators, who select the image that best matches the target category when a suitable candidate is available. If none of the five retrieved images is suitable for a category, that category is discarded. Table 9 summarizes the number of category labels before and after this filtering process.

After filtering, we randomly combine several retained categories within each domain to form image-generation specifications. For daily objects, animals, plants, and electronics, each generated image contains categories sampled from a range of 3 to 5. For cars, we use a smaller range of 2 to 4 categories, since fine-grained car categories often require more visual space to remain identifiable.

Model	Access	Size
InstructBLIP-Vicuna-7B (Dai et al., 2023)	Local HF checkpoint	7B
InternVL3.5-8B (Zhu et al., 2025)	Local HF checkpoint	8B
Qwen2.5-VL-7B (Bai et al., 2025b)	Local HF checkpoint	7B
Qwen3-VL-8B-Instruct (Bai et al., 2025a)	Local HF checkpoint	8B
Kimi-K2.6 (Moonshot AI, 2026)	API	–
GLM-4.6V (Hong et al., 2025)	API	–
GPT-5.4 (OpenAI, 2026)	API	–
Gemini-3.1-Flash-Lite (Google DeepMind, 2026)	API	–

Table 7: Evaluated MLLMs. The access column indicates whether each model is evaluated by loading local HuggingFace weights or through an API. For API-accessed models, we leave the size unspecified.

Prompt style	Prompt
Neutral	Describe the image. Respond in English only.
Conservative	Describe the image factually and refrain from guessing unobserved details. Respond in English only.
Aggressive	Describe the image in as much detail as possible. Respond in English only.

Table 8: Evaluation prompts used for the three prompt styles.

Domain	Images	Labels	Objects	Avg. obj/img
Daily objects	250	496	990	3.96
Animals	50	104	194	3.88
Plants	50	63	204	4.08
Cars	75	104	195	2.60
Electronics	100	208	403	4.03
Total	525	975	1986	3.78

Table 10: Domain-level statistics of the auxiliary training set.

Domain	Initial labels	Retained labels
Daily objects	600	496
Animals	150	104
Plants	100	63
Electronics	250	208
Cars	150	104
Total	1250	975

Table 9: Category-label filtering statistics for the auxiliary training set.

We then use Gemini-3.1-Flash-Image-Preview to synthesize images from these category combinations. After generation, we use GPT-5.4 to produce structured annotations for the synthesized images, including entity-level category paths and visual attributes. The generated image–annotation pairs are then checked for validity.

The final auxiliary training set contains 525 images and 1986 annotated object instances. Table 10 summarizes the per-domain statistics.

The auxiliary training set is constructed using textual category labels from the GRANFACT domains, while all training images are kept image-disjoint from the GRANFACT evaluation set.

Figure 12 visualizes the domain distribution, object-count distribution, and category-depth distribution of the auxiliary training set.

Overall, the auxiliary training set provides a compact source of structured fine-grained supervision for preference construction. Its scale is deliberately smaller than typical large-scale instruction-tuning datasets, reflecting a low-resource setting where only a limited number of fine-grained structured annotations are available.

D.5 Reliability-Guided Semantic Rollback Prompts

Reliability-Guided Semantic Rollback (RSR) is used to construct reliability-oriented preference pairs. Given an original model response and the diagnostic results from the evaluation pipeline, RSR revises unsupported or overly specific category claims while preserving the supported content of the response as much as possible.

RSR uses the image-specific GT category hierarchy to decide whether a predicted category should be kept, deleted, or rolled back. For each predicted category, we provide the LLM judge with the GT hierarchy of the current image. The judge first determines the deepest GT-supported node that the prediction can match. It then checks whether the prediction is consistent with the child nodes under that matched node. This second check is important because an over-specific prediction may share a coarse ancestor with the GT hierarchy while still making a mutually exclusive fine-grained claim.

Based on this hierarchy-aware judgment, RSR generates edit commands using the following rules. If a predicted category cannot match any node

in the GT hierarchy, the corresponding claim is marked for deletion. If the prediction matches a non-leaf node but contains a fine-grained claim that conflicts with the child nodes under that node, the claim is rolled back to the matched node. Otherwise, the prediction is kept unchanged. The resulting edit commands are then applied to the original response by a modifier model.

The modifier receives the original response and the edit instructions, and directly returns the revised English description. It is instructed to apply only the specified edits, preserve unrelated content, and avoid introducing new visual claims.

When RSR produces a revised response, we pair it with the original response to form a reliability preference pair. The revised response is treated as preferred because it preserves the supported content of the original response while reducing unsupported or overly specific category claims.

D.6 Preference Pair and Margin Construction

This subsection provides a brief recap of how the training preference pairs are instantiated, together with illustrative examples. As described in Section 5.2, we construct two complementary preference sets: reliability preferences $\mathcal{D}_{\text{rel}}$ and granularity preferences $\mathcal{D}_{\text{gran}}$.

For each training image I, we sample K candidate responses $\mathcal{Y}_{\text{orig}}(I) = \{y_{\text{orig}}^{(k)}\}_{k=1}^{K}$ from the base model, and then apply Reliability-Guided Semantic Rollback (RSR) to obtain the rectified response pool $\mathcal{Y}_{\text{rect}}(I) = \{\text{RSR}(y_{\text{orig}}^{(k)})\}_{k=1}^{K}$. All preference pairs are constructed within the same image, so that the comparison reflects response quality rather than image difficulty.

Reliability preferences. Reliability preferences compare an original response with its RSR-rectified version. When RSR modifies a response by deleting an unsupported category claim or rolling back an over-specific claim to a supported coarser category, we form a preference pair $(I, y_+, y_-) = (I, y_{\text{rect}}^{(k)}, y_{\text{orig}}^{(k)})$. The rectified response is treated as preferred because it preserves the supported content of the original response while reducing unsupported claims. For such pairs, the margin is computed from the improvement in granularity-neutral precision, following Section 5.3.

Granularity preferences. Granularity preferences are constructed within the rectified response pool of the same image. Specifically, we compare pairs (y_+, y_-) drawn from $\mathcal{Y}_{\text{rect}}(I)$ and retain a pair only when $\text{F1}_{\text{gran}}(I, y_+) > \text{F1}_{\text{gran}}(I, y_-) + \tau$. This encourages the model to prefer more informative responses after reliability has already been improved by RSR. For these pairs, the margin is computed from the normalized F1_{gran} gap.

Margins. Each preference pair is assigned an instance-specific margin $m = \gamma + \alpha\delta$, where $\delta \in [0, 1]$ is the normalized metric gap of that pair. The gap definition depends on the preference type: reliability pairs use the improvement in P, while granularity pairs use the improvement in F1_{gran}. In practice, we use larger margin hyperparameters for reliability preferences than for granularity preferences, so that the overall training objective remains reliability-first while still rewarding supported fine-grained descriptions.

Figure 15 shows an example of a reliability preference pair, including the original response, the RSR-rectified response, their metric scores, and the resulting margin. Figure 16 shows an example of a granularity preference pair, where both responses are reliable, but the preferred one receives a higher granularity-aware score and thus a positive margin.

D.7 Training Hyperparameters

For the model budget, RP-DPO is trained on Qwen3-VL-8B-Instruct, which has approximately 8B parameters. Training is conducted with LoRA, so only a small subset of adapter parameters is updated while the base model weights remain frozen. All training runs are performed on 8 NVIDIA GeForce RTX 4090D GPUs. The full RP-DPO training run takes approximately **6** wall-clock hours, corresponding to approximately **48**× GPU-hours.

We train our method starting from Qwen3-VL-8B-Instruct as the base policy model. The reference model in DPO is initialized from the same checkpoint and kept frozen unless otherwise specified. Across response extraction, assignment-score computation, and RSR, we use Qwen3.5-27B as the LLM judge with temperature set to 0 for reproducibility.

We use MS-Swift as the training framework and implement the RP-DPO objective with our custom `RPDPOTrainer`. The main training settings are summarized in Table 11. Unless otherwise specified, we use the final checkpoint after training for evaluation.

Hyperparameter	Value
Base policy model	Qwen3-VL-8B-Instruct
Reference model	Frozen initial policy
LLM judge	Qwen3.5-27B
Judge temperature	0
Training framework	MS-Swift
Trainer	Custom `RPDPOTrainer`
λ	0.5
γ_{rel}	0.1
α_{rel}	0.5
γ_{gran}	0
α_{gran}	0.3
DPO β	0.1
Pair filtering threshold τ	0.002
Learning rate	1×10^{-4}
Per-device train batch size	1
Gradient accumulation steps	16
Training steps	33
Precision	bf16
Model size	8B parameters
Hardware	8 × NVIDIA GeForce RTX 4090D GPUs
GPU memory	24GB per GPU
Training wall-clock time	**6** hours
Training compute budget	**48** GPU-hours
LoRA rank	32
LoRA alpha	32
LoRA target modules	-all-linear

Table 11: Training settings for RP-DPO.

E Additional Experimental Results

E.1 Qualitative Examples and Error Analysis

Figure 17 presents the original responses generated by Gemini-3.1-Flash under conservative and aggressive prompts, and Figure 18 further visualizes how our evaluator scores them. For each response, we first parse object-level predictions and then compute an optimal global matching to the ground-truth (GT) objects. For a sample z, we report both the matching size M_z, i.e., the number of matched prediction–GT pairs under the optimal assignment, and the total granularity score $M_{z,\text{gran}}$, i.e., the sum of the edge-level granularity scores over all matched pairs.

Under the conservative prompt, the parsed predictions are *smartphone*, *Samsung smartphone*, *Samsung smartphone*, and *foldable smartphone*. All four predictions can be matched to GT objects, giving $M_z = 4$. However, these matches remain at relatively coarse semantic levels, resulting in a lower total granularity score $M_{z,\text{gran}} = 1.6123$.

Under the aggressive prompt, the parsed predictions become *Galaxy Z Fold smartphone*, *Samsung smartphone*, *Galaxy S24 Ultra smartphone*, and *Galaxy Z Flip smartphone*. The first and fourth predictions are more specific and receive substantially higher granularity scores, which increases the total granularity score to $M_{z,\text{gran}} = 2.0459$. However, the third prediction is an unsupported fine-grained claim: it identifies the true *Samsung Galaxy S23 Ultra smartphone* as a *Galaxy S24 Ultra smartphone*. As a result, this prediction remains unmatched, the corresponding GT object is missed, and the matching size drops to $M_z = 3$.

This example illustrates the reliability–granularity trade-off captured by our evaluation. Aggressive prompting can improve semantic specificity for matched objects, but it may also introduce unsupported fine-grained predictions that reduce object coverage under global matching. By reporting both M_z and $M_{z,\text{gran}}$, our evaluator makes this trade-off explicit at the sample level.

F Artifact Use, License, and Intended Use

Artifacts used and created. This work uses publicly available and API-based multimodal large language models for research evaluation and, where applicable, model training. We cite the creators of the main external artifacts used or built upon, including pretrained MLLMs and Direct Preference Optimization. Prior datasets and benchmarks discussed for comparison are cited as related work.

We introduce GRANFACT, including expert-verified image annotations, a hierarchy-aware evaluation protocol, and evaluation code. The benchmark images are collected from public web sources and real-world photographs. For images collected from public sources, redistribution will follow the corresponding source licenses or terms of use. When redistribution of raw images is not permitted or cannot be verified, we will release only metadata, annotations, and evaluation scripts, or provide instructions for reconstructing the benchmark where appropriate. The annotations and evaluation code will be released under research-friendly licenses, such as CC BY-NC 4.0 for annotations and MIT License for evaluation code, subject to compatibility with the licenses and terms of the underlying resources.

Privacy and offensive content. GRANFACT is designed to evaluate object-level fine-grained visual description rather than person identification. During data collection and annotation, we screen images to avoid content that names or uniquely identifies individual people, such as faces, personal documents, account names, or other personally identifying information. We also filter out offensive, hateful, sexually explicit, or otherwise unsafe content. The released annotations describe object categories, quantities, and visual attributes, and do not include personal information. If any problem-

atic content is later identified, we will remove or replace the corresponding example.

Documentation and statistics. We document the construction and coverage of GRANFACT, including image sources, visual domains, annotation protocol, category hierarchy, object-level annotations, and evaluation metrics. We also report dataset statistics, including the number of evaluation images, domain distribution, number of annotated objects per image, and granularity depth. The auxiliary training set used for RDPO is disjoint from the GRANFACT evaluation set, and its construction is described separately in the appendix.

Intended use. GRANFACT and the associated evaluation protocol are intended for research on reliable fine-grained multimodal generation and evaluation. They are designed to analyze whether MLLMs can generate visually grounded descriptions at appropriate semantic granularities. They should not be used as guarantees of correctness, nor for high-stakes decision making, privacy-sensitive identification, surveillance, or other applications where unsupported fine-grained predictions could cause harm. Users of the artifacts are responsible for complying with the licenses and terms of use of any underlying images, models, and APIs.

Response Parsing Prompt

You are an expert extraction assistant. Given an MLLM response, extract all in-scope primary physical objects described in the text into exactly one JSON array.

Scope

Extract primary physical objects that belong to the target domain of the image. Exclude generic background, environment, surface, lighting, UI elements, and low-information context objects unless the response itself focuses on them. Minor accessories, holders, connectors, supports, or object parts should usually be kept as attributes or omitted, unless they are described as main objects.

Extraction Rules

- Output exactly one JSON array and no extra text.
- Each extracted object must contain a category field and an `attributes` field.
- Use the most specific category explicitly supported by the response. Every word in the category should be grounded in the text.
- Keep visual details such as color, material, texture, size, shape, position, orientation, state, and condition in `attributes`, unless they are part of an identity-bearing object name.
- Put quantity and position inside `attributes`, not as top-level fields. If quantity is stated or reasonably implied, store it in `attributes.number`.
- Use an exact integer string for a specific quantity; use `"1"` for one countable object instance; use `"uncertain"` for approximate, plural-but-unspecified, or unclear quantities.

- Do not merge objects with different explicit labels, series, specifications, capacities, or form factors.

Category-Level Matching Judge Prompt

You are a category-level matching judge for fine-grained visual evaluation. Given a predicted category and the coarse-to-fine category path of a GT entity, determine the finest valid GT level that the prediction can match.

Input

- **Predicted category:** `{predicted_category}`
- **GT category path:** `{coarse_to_fine_category_path}`

Decision rules

- Match the prediction to the finest GT level that is explicitly or canonically supported by the predicted category.
- Treat synonyms, paraphrases, aliases, and common naming variants as valid matches when they refer to the same semantic scope.
- Missing specificity is allowed. If the prediction is correct but less specific than the GT path, match it to the corresponding coarser GT level.
- Ignore non-identity visual modifiers such as color, material, texture, size, pose, location, or condition, unless they are part of a recognized identity-bearing category name.
- Do not match a prediction to a coarser GT level if the prediction explicitly asserts a mutually exclusive fine-grained identity, such as a different species, breed, brand, model, series, generation, flavor, version, character, or landmark.
- Do not infer hidden categories that are not supported by the GT path.

Output

Return only a JSON object in the following format:

```
{
  "matched_level": 0,
  "matched_category": null,
  "reason": "brief explanation"
}
```

Use `matched_level = 0` and `matched_category = null` if the prediction is incompatible with the GT entity. Otherwise, `matched_level` should be the 1-indexed level in the GT category path, and `matched_category` should be the matched GT category at that level.

Examples

- GT path: `smartphone -> iPhone -> iPhone 15 Pro Max`. Prediction: `iPhone`. Match to `iPhone`.
- GT path: `smartphone -> iPhone -> iPhone 15 Pro Max`. Prediction: `iPhone 15 Pro Max`. Match to `iPhone 15 Pro Max`.
- GT path: `smartphone -> iPhone -> iPhone 15 Pro Max`. Prediction: `iPhone 15 Pro`. Return incompatible, because the prediction asserts a different fine-grained model.

Figure 10: LLM judge prompt for determining the matched category granularity. Level ℓ_{ij} between a prediction entity and a GT entity.

Attribute Coverage Judge Prompt

You are a pair-level attribute coverage judge for fine-grained visual evaluation. Given one predicted object and several category-legal GT candidates, identify which GT attribute facts are explicitly covered by the predicted attributes.

Input

- **Predicted object:** `{predicted_object}`
- **Category-legal GT candidates:** `{gt_candidates}`

Decision rules

- Category legality has already been decided upstream. Do not perform category matching.
- Do not perform global assignment, reason about competition with other predictions, or do quantity allocation.
- For each candidate GT, identify which GT attribute facts excluding number are explicitly covered by the predicted attributes.
- Mark whether there is an explicit contradiction in attributes.
- Be literal and conservative. Do not infer hidden facts.

Output

Return only one JSON object with the following structure:

```
{
  "candidates": {
    "g3": {
      "matched_gt_fact_indices": [0, 2],
      "has_contradiction": false
    },
    "g5": {
      "matched_gt_fact_indices": [],
      "has_contradiction": true
    }
  }
}
```

Figure 11: LLM judge prompt for identifying GT attribute facts covered by a predicted entity for category-legal GT candidates.

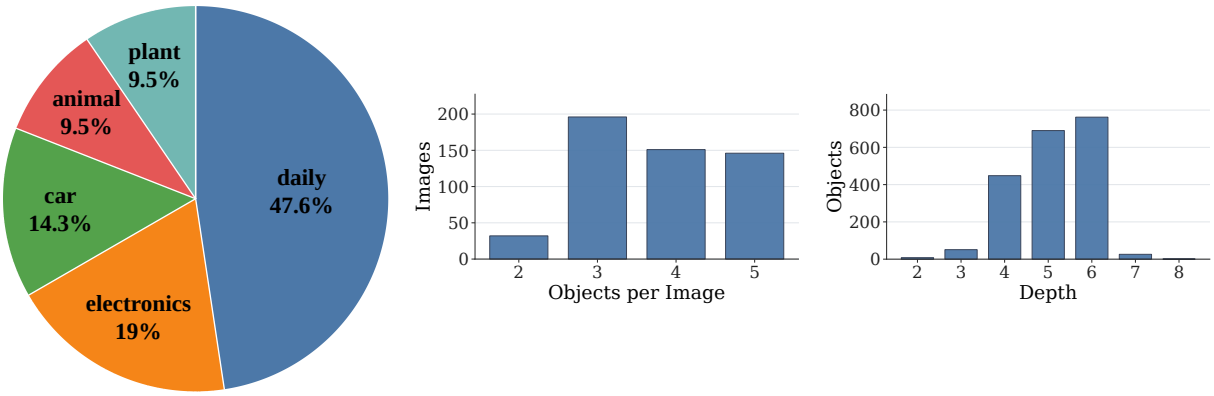

Figure 12: Statistics of the auxiliary training set. Left: domain distribution. Middle: number of annotated objects per image. Right: distribution of category-path depths.

RSR Edit Command Prompt

Role. You are a hierarchy-aware reliability judge. Given predicted category claims from a model response and the GT category hierarchy of the current image, decide whether each claim should be kept, deleted, or rolled back.

Input. The input contains: (1) predicted category texts extracted from the model response, and (2) the image-specific GT category hierarchy.

Decision rules. For each predicted category, find the deepest GT-supported hierarchy node that the prediction can match. If no GT-supported node can be matched, mark the claim as `delete`. If the prediction matches a non-leaf node but makes a fine-grained claim that conflicts with the child nodes under that node, mark the claim as `rollback` and roll it back to the matched node. Otherwise, mark the claim as `keep`. Missing specificity is not a conflict: a coarse but correct prediction should be kept. Ordinary visual modifiers such as color, material, size, pose, or position should not be treated as category conflicts unless they are part of an identity-bearing category name.

Output. Return only a JSON array. Each item should correspond to one predicted category claim:

```
[
  {
    "predicted_category": "...",
    "action": "keep | delete | rollback",
    "matched_node": "... or null",
    "rollback_to": "... or null",
    "reason": "brief explanation"
  }
]
```

For `keep`, set `rollback_to` to `null`. For `delete`, set both `matched_node` and `rollback_to` to `null`. For `rollback`, set `rollback_to` to the supported GT hierarchy node that should replace the unsupported fine-grained claim.

Figure 13: Prompt used for generating edit commands in Reliability-Guided Semantic Rollback.

RSR Modifier Prompt

Role. You are a careful revision assistant for image descriptions. Your task is to minimally edit an existing English image description according to explicit object-level edit instructions.

Input. The input is a JSON object containing the original description and a list of edit instructions:

```
{
  "original_description": "...",
  "edit_instructions": [
    {
      "action": "rollback | delete",
      "source_category": "...",
      "target_category": "... or null"
    }
  ]
}
```

Instructions. For `action=rollback`, rewrite mentions of `source_category` as `target_category` and remove incompatible identity-specific details. For `action=delete`, remove sentences or clauses describing the hallucinated `source_category`. If a hallucinated object is intertwined with a sentence about real objects, edit only the hallucinated part and keep the real-object content. Preserve all unrelated content as much as possible. Do not add new objects, attributes, counts, or visual details. Keep the final description fluent, coherent, and in English.

Output. Return only the final revised English description, with no explanation, no markdown, and no quotes around the whole answer. Do not mention the edit instructions or the words `hallucination`, `source_category`, or `target_category` in a meta way.

Figure 14: Prompt used for applying edit instructions and producing the RSR-revised response.

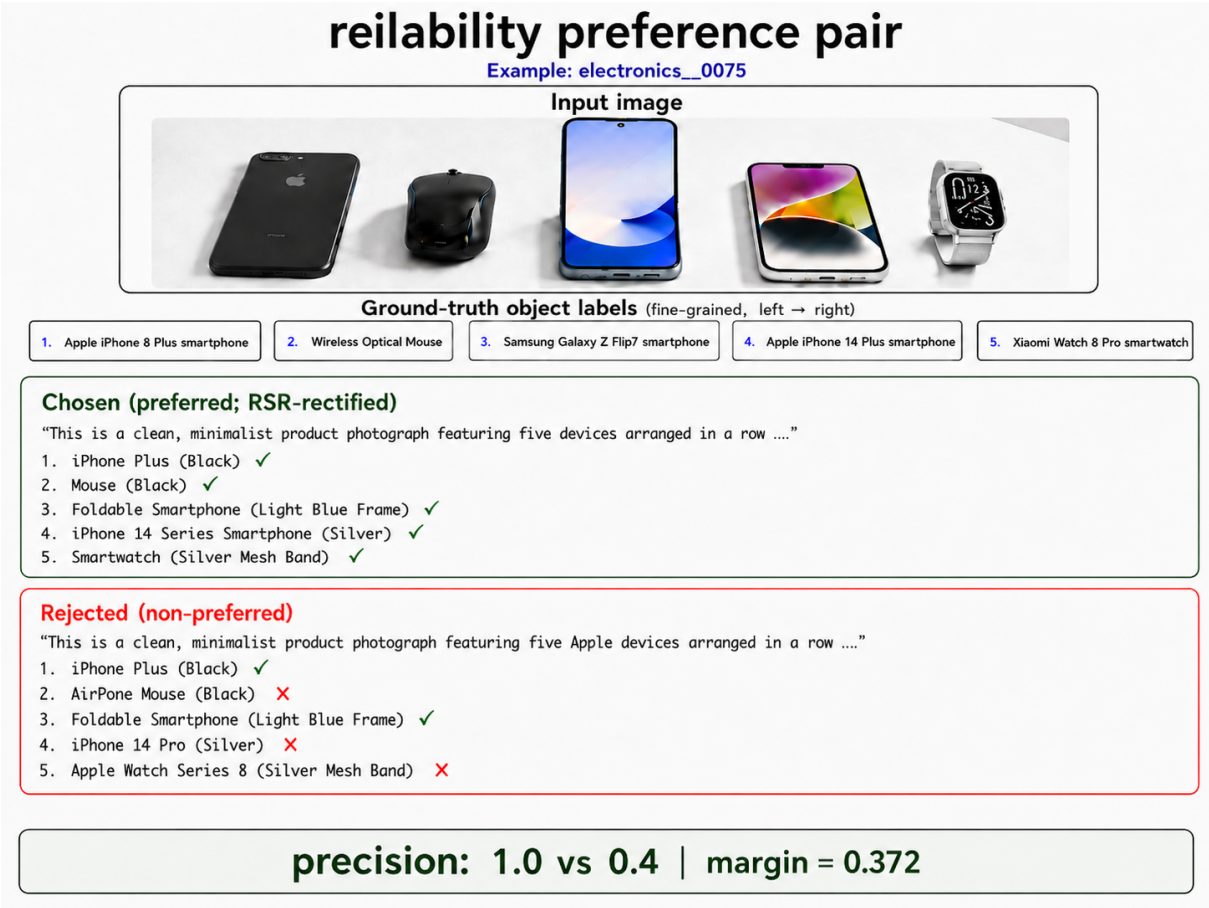

Figure 15: Example of a reliability preference pair. The preferred response is the RSR-rectified response, while the dispreferred response is the original response. The figure also shows the corresponding evaluation scores and the margin computed from the precision gap.

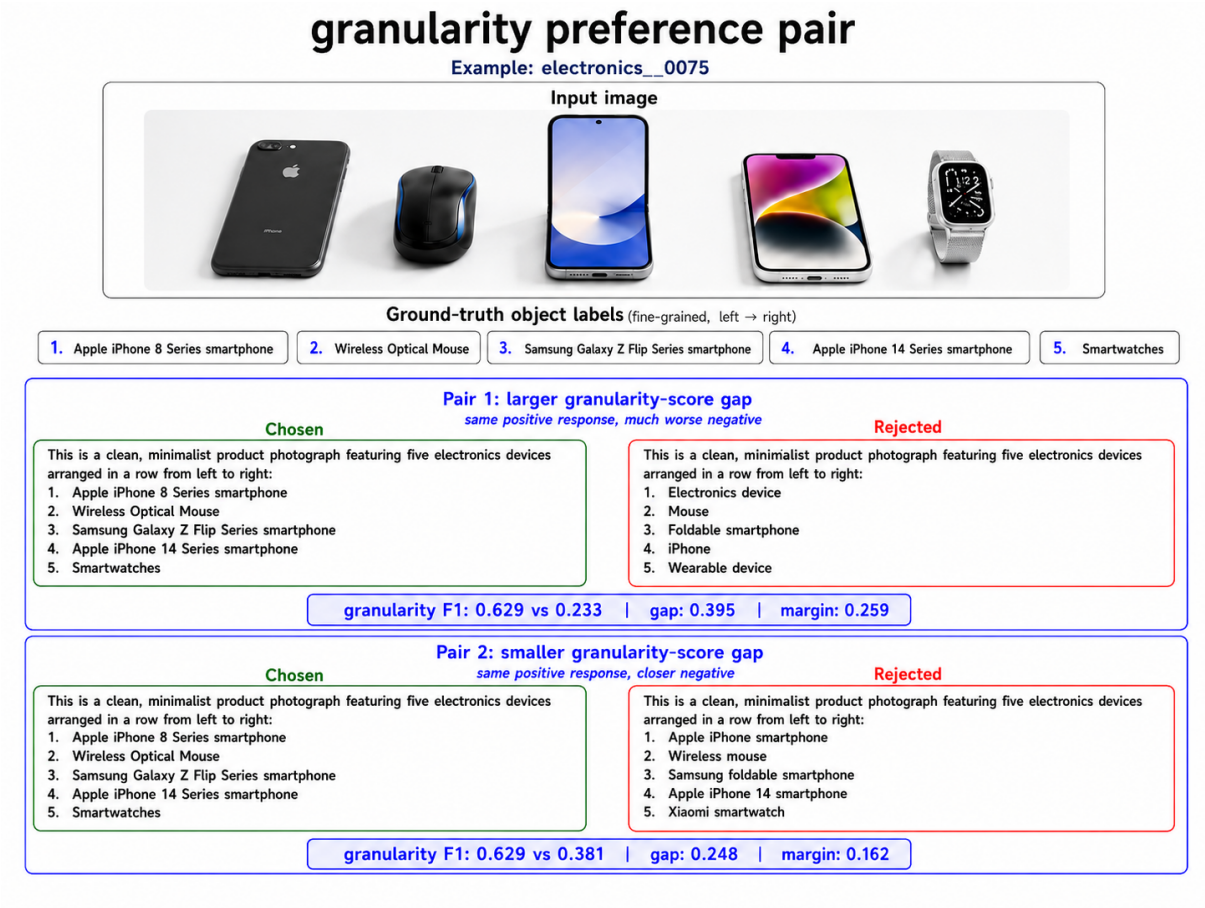

Figure 16: Example of a granularity preference pair. Both responses are drawn from the RSR-rectified response pool of the same image. The preferred response is selected because it achieves a higher granularity-aware score, and the figure shows the corresponding metric gap and margin.

Gemini-3.1-Flash under the conservative prompt

The image displays four smartphones arranged in a row on a white surface, viewed from the back.
From left to right:

1. **First phone:** A tall, narrow, light-colored device with a vertical camera module containing three lenses.
2. **Second phone:** A light-colored device with a vertical camera module containing two lenses and a flash. The word "SAMSUNG" is visible near the bottom.
3. **Third phone:** A light-colored device with a camera array consisting of five circular elements arranged in a "P" shape. The word "SAMSUNG" is faintly visible near the bottom.
4. **Fourth phone:** A foldable device shown in its closed state. It has a dark upper section containing two camera lenses and a flash, and a light-colored lower section.

Gemini-3.1-Flash under the aggressive prompt

This high-angle, eye-level shot displays four Samsung smartphones arranged in a row on a plain, light-colored surface. The phones are positioned vertically, showing their back panels.
Starting from the left:

1. **First Phone:** This is a tall, narrow device, likely a Galaxy Z Fold model. It has a matte, light-gray finish. On the left side, there is a vertical camera module containing three circular lenses stacked one above the other.
2. **Second Phone:** This device has a smooth, metallic, champagne-gold finish. It features a vertical camera housing on the top left with two large, prominent circular lenses and a small flash below them. The "SAMSUNG" logo is faintly visible in the lower center of the back panel.
3. **Third Phone:** This appears to be a Galaxy S24 Ultra or similar model. It has a similar champagne-gold, matte finish. The camera array is distinct, featuring five circular elements: three large lenses arranged vertically on the left, and two smaller circular components, a lens and a flash/sensor, to their right. The "SAMSUNG" logo is subtly embossed near the bottom.
4. **Fourth Phone:** This is a foldable device, likely a Galaxy Z Flip model, shown in its closed state. The top half of the back is a glossy black panel containing two circular camera lenses and a small flash. The bottom half is a matte, muted-brown or taupe-colored panel. The phone has a metallic frame visible along the edges.

The lighting is even and soft, casting minimal shadows, which highlights the different textures and finishes of the devices. The background is a clean, neutral white or off-white surface.

Figure 17: Qualitative responses of Gemini-3.1-Flash under conservative and aggressive prompts.

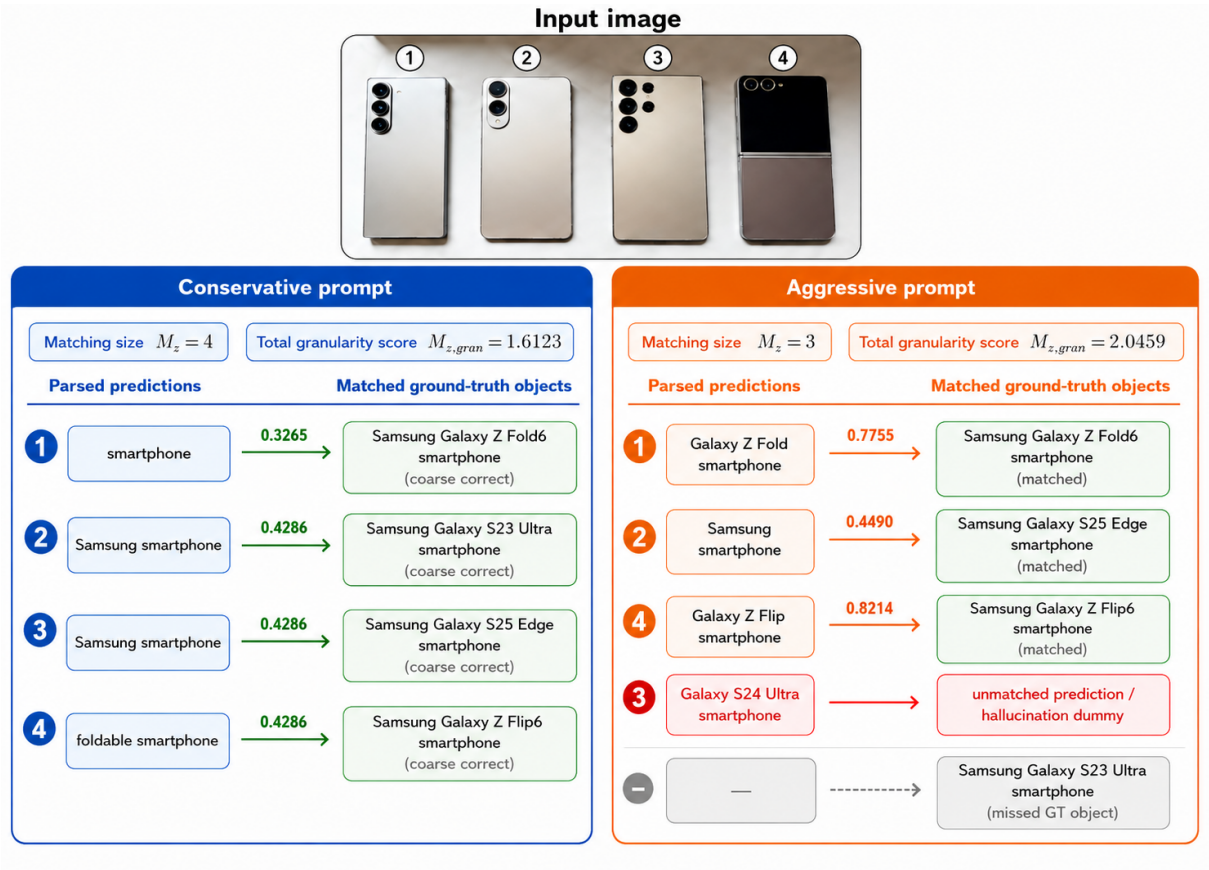

Figure 18: Evaluation outcomes for the qualitative example in Figure 17. The conservative response matches all four GT objects ($M_z = 4$) but at coarse semantic levels ($M_{z,\text{gran}} = 1.6123$). The aggressive response achieves finer matched descriptions ($M_{z,\text{gran}} = 2.0459$) but introduces an unsupported fine-grained claim, reducing the reliablility score to $M_z = 3$.