REVEALER: Reinforcement-Guided Visual Reasoning for Element-Level Text-Image Alignment Evaluation

Anonymous ACL submission

Abstract

Evaluating the alignment between textual prompts and generated images is critical for ensuring the reliability and usability of text-to-image (T2I) models. However, most existing evaluation methods rely on coarse-grained metrics or static Question Answering (QA) pipelines, which lack fine-grained interpretability and struggle to reflect human preferences. To address this, we propose **REVEALER**, a reinforcement-guided visual reasoning framework for element-level text-to-image alignment evaluation. Adopting a structured "grounding–reasoning–conclusion" paradigm, our method enables Multimodal Large Language Models (MLLMs) to explicitly localize semantic elements and derive interpretable alignment judgments. We optimize the model via Group Relative Policy Optimization (GRPO) using a multi-dimensional reward function that targets format compliance, localization precision, and alignment accuracy. Extensive experiments confirm that REVEALER achieves state-of-the-art results across four benchmarks. Notably, on EvalMuse-40K, it surpasses the strong proprietary Gemini 3 Pro and Training-based baselines with absolute accuracy gains of **+4.0%** and **+13.1%**, respectively. Ablation studies further demonstrate the efficacy of our method, contributing a cumulative **19.4%** improvement over the base model. Code: https://anonymous.4open.science/r/F698/

1 Introduction

Text-to-image (T2I) models (Podell et al., 2023) such as DALL·E (Ramesh et al., 2021), Stable Diffusion (Rombach et al., 2022; Esser et al., 2024), and Imagen (Saharia et al., 2022) have made significant strides in generating visually appealing and semantically rich images from natural language prompts. With the widespread adoption of T2I models, ensuring that the generated image faithfully aligns with the semantics of the input text becomes increasingly critical, which is known as the task of *text-image alignment evaluation*.

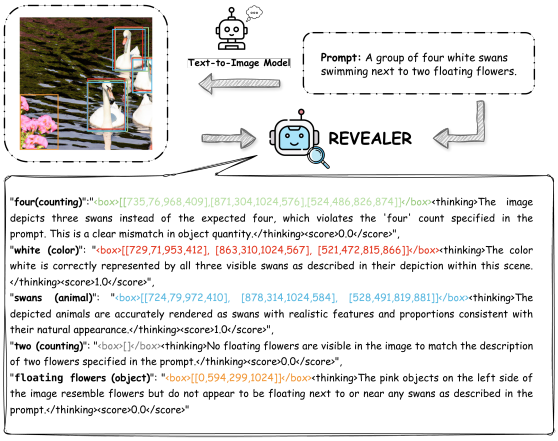


Figure 1: **REVEALER** performs element-level text-to-image alignment evaluation via structured visual reasoning, following a grounding–reasoning–conclusion paradigm.

Early text-image alignment evaluation methods (Heusel et al., 2018; Hessel et al., 2022; Salimans et al., 2016) rely on coarse-grained metrics that collapse rich semantic structures into single scalar scores (e.g., CLIPScore (Hessel et al., 2022)), but they lack interpretability and are often insensitive to fine-grained mismatches, such as object count, attributes, and spatial composition. To improve the interpretability of CLIPScore, VIEScore (Ku et al., 2024) proposes leveraging multimodal large language models (MLLMs) (Liu et al., 2023; Bai et al., 2023; Dai et al., 2023) to generate natural language rationales alongside alignment scores. To facilitate fine-grained alignment evaluation, question-answering (QA)-based approaches (Huang et al., 2025; Lu et al., 2023), such as TIFA (Hu et al., 2023) and VQ^2 (Yarom et al., 2023), employ off-the-shelf large language models (LLMs) to generate multiple verifiable questions from the input prompts, with each question targeting a distinct facet of alignment evaluation. However, due to

their reliance on predefined question templates, these methods often fail to generate questions that adequately assess the alignment of all elements in the prompt, especially in complex cases. Moreover, most MLLM-based QA alignment evaluation methods rely solely on prompt engineering without dedicated supervision, resulting in suboptimal evaluation performance. To address these issues, EvalMuse-40K (Han et al., 2024) introduces a large-scale benchmark featuring element-level binary annotations (e.g., objects, attributes, locations), providing rich supervision for fine-grained alignment training and evaluation. However, its annotations are often treated as classification tasks, lacking interpretable reasoning paths. Recently, UnifiedReward-R1 (Wang et al., 2025a) has explored reinforcement learning to enable chain-of-thought-style text-image alignment score prediction by incorporating rule-based reward signals. However, UnifiedReward-R1 only provides an overall alignment score for each evaluated dimension, and lacks the capability to explicitly determine whether specific objects or elements are correctly generated according to the input prompt.

To address the aforementioned limitations, we propose **REVEALER**, a reinforcement-guided visual reasoning framework for element-level text-to-image alignment evaluation. As illustrated in Figure 1, REVEALER operates through a three-stage framework comprising grounding, reasoning, and conclusion, which emulates human-like analysis in text-image alignment evaluation. At the first stage, the visual reasoning grounds each element of the prompt to specific regions within the generated images, thereby providing essential contextual information for alignment reasoning. Here, the elements are derived by decomposing the input prompt into fine-grained semantic units, which follows the TIFA taxonomy categorization (e.g., object, attribute, activity, etc.). At the second stage, a free-form natural language explanation is produced to evaluate the alignment between the grounded visual content and the corresponding element in the prompt. Finally, an element-level alignment score is derived by comprehending the information obtained from the grounding and reasoning stages. This interleaved visual-textual reasoning process significantly improves the interpretability of the evaluation metric, while simultaneously offering dense supervision signals for model training.

To equip the MLLM with the capability to follow the three-stage visual reasoning paradigm, we first fine-tune it on automatically curated visual reasoning trajectories. Subsequently, a reinforcement learning (RL) phase—implemented via Group Relative Policy Optimization (GRPO) (Shao et al., 2024)—is employed to bolster the model's reasoning capabilities. Specifically, we design a comprehensive rule-based reward function to leverage the rich supervision signals intrinsic to all three stages. To facilitate this training recipe, we propose an automated data curation pipeline that synthesizes training trajectories by synergizing an expert vision model with general-purpose LLMs.

Extensive experiments across four benchmarks demonstrate that REVEALER achieves state-of-the-art performance. Specifically, our method yields substantial accuracy gains relative to the Training-based baseline, achieving increases of **+13.1%** on EvalMuse-40K, **+9.4%** on RichHF, **+6.3%** on MHaluBench, and **+6.5%** on GenAI-Bench. Notably, it surpasses the strong proprietary model, Gemini 3 Pro, by a margin of **+4.0%** on EvalMuse-40K. Ablation studies further validate the efficacy of our framework components, showing a cumulative performance boost of **+19.4%** over the base model, while subsequent analyses confirm that explicit visual reasoning enhances both fine-grained alignment accuracy and interpretability.

2 Related Work

This section provides a brief review of related work.

Automated Methods and Metrics for Text-Image Alignment Evaluation. Early metrics such as (Hessel et al., 2022; Li et al., 2023; Kirstain et al., 2023) evaluate text-image alignment via cosine similarity in embedding space. While computationally efficient, these approaches lack sensitivity to fine-grained mismatches. To improve interpretability, structured evaluation methods such as TIFA (Hu et al., 2023) and VQ^2 (Yarom et al., 2023) convert prompts into QA or NLI tasks, though their performance depends heavily on hand-crafted templates. More recent efforts introduce stronger compositional reasoning: VIEScore (Ku et al., 2024) uses instruction-following MLLMs to generate alignment scores with natural language rationales; DSG (Cho et al., 2024) leverages semantic scene graphs for robustness; And VQAScore (Lin et al., 2024) decomposes prompts into atomic QA sub-tasks for modular evaluation. FGA-BLIP2 (Han et al., 2024) fine-tunes models for element-level alignment scoring, while PN-VQA (Han et al.,

2024) adopts a prompt-based querying strategy without fine-tuning. A recent task-decomposed framework (Tu et al., 2024) further enhances interpretability and robustness by combining modular pipelines with multi-perspective metrics. In parallel, MLLM-based methods (Tan et al., 2024) directly predict alignment scores through supervised finetuning on human-aligned data.

Reinforcement Learning for Visual Reasoning and Evaluation. Reinforcement learning (RL) has been used to enhance alignment evaluation, as in T2I-Eval-R1 (Ma et al., 2025), UM-CoT-RM (Wang et al., 2025a), Unified Hallucination Detection (Chen et al., 2024), UnifiedReward (Wang et al., 2025b) and Vision-R1 (Zhan et al., 2025), which aim to enhance alignment consistency in visual content and improving interpretability through reasoning chains. RL also improves visual reasoning: DeepEyes (Zheng et al., 2025) and OpenThinkIMG (Su et al., 2025) train agents for spatial reasoning, and Q-Insight (Li et al., 2025) applies reinforcement learning to train visual agents for interpretable image quality assessment. ViLaSR (Wu et al., 2025) reinforces geometric understanding, and works like Thinking with Generated Images (Chern et al., 2025), Chain-of-Focus (Zhang et al., 2025), and UniVG-R1 (Bai et al., 2025) explore internal reasoning via sketching, zooming, or CoT-based image generation. These efforts demonstrate how sequential visual reasoning enhances robustness and explainability.

3 Methodology

In this section, we first introduce the visual reasoning process for element-level text-image alignment evaluation §3.1. We then describe the training dataset curation procedure §3.2, followed by a detailed illustration of the two-stage training methodology §3.3. The overall methodology is illustrated in Figure 2.

3.1 Visual Reasoning for Element-Level Text-Image Alignment Evaluation

Despite recent advances (Zheng et al., 2025; Su et al., 2025; Li et al., 2025), existing approaches to T2I alignment evaluation still struggle with accurately assessing element-level alignment between textual descriptions and generated images. Inspired by the human-like alignment analysis process, which follows a three-stage chain-of-thought "grounding—reasoning—conclusion", we propose visual reasoning guided element-level text-image alignment evaluation via reinforcement learning.

Specifically, the visual reasoning process unfolds in three stages, each corresponding to a structured component in the reasoning trajectory. In the **grounding** stage, the sequence begins with a special token <box>, followed by a predicted bounding box list that localizes a semantic element from the input prompt within the generated image. Next, in the **reasoning** stage, the <thinking> token precedes a free-form natural language explanation that evaluates the semantic alignment between the visual content in the localized region and the corresponding element in the prompt. Finally, in the **conclusion** stage, the sequence begins with the <score> token followed by a scalar alignment score $s \in [0, 1]$, where the scalar magnitude quantifies the degree of visual-semantic consistency, with higher values signifying superior alignment.

This three-stage visual reasoning alignment evaluation offers several notable advantages. First, by explicitly localizing specific semantic elements within the generated image, the grounding stage facilitates more precise visual-textual alignment and provides essential contextual information for subsequent reasoning. Second, the intermediate natural language rationales generated in the the reasoning stage enhance the interpretability of the final alignment score. Lastly, this staged visual reasoning yields rich supervision signals for both training and evaluation, as will be further detailed in the following sections.

Visual Reasoning Trajectory Curation. To support the aforementioned visual reasoning training, we propose an automated method for curating such visual reasoning trajectory, which combines the visual grounding capability of an expert model and the reasoning ability of proprietary LLMs. The overall curation process is shown in Figure 2 (a).

The visual reasoning dataset is derived from the training split of EvalMuse-40K. EvalMuse-40K is a large-scale benchmark for text-to-image alignment evaluation, which contains 40K image–prompt pairs with element-level binary annotations.

Specifically, let $(\mathcal{I}, \mathcal{P}, \{\langle e_i, a_i \rangle\}_{i=1}^{N})$ denote a data point in EvalMuse-40K, where $\mathcal{I}$ is the generated image and $\mathcal{P}$ is the input prompt. $\{\langle e_i, a_i \rangle\}_{i=1}^{N}$ corresponds to the set of the element-level annotations (e.g., objects, attributes, locations), where e_i denotes an element, and a_i denotes the binary answer. The visual reasoning trajectory for each data point is constructed as follows. First, for

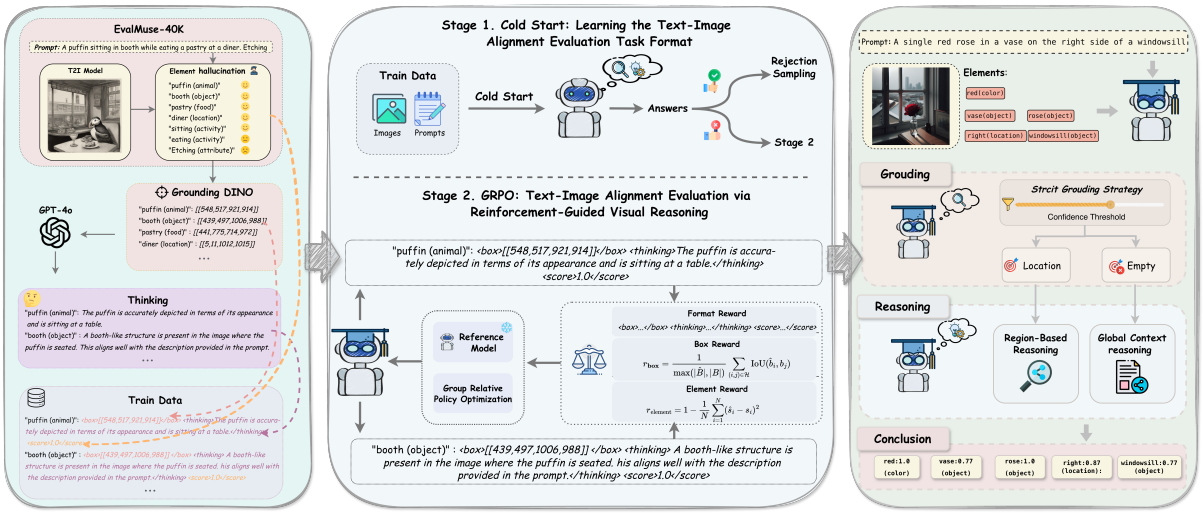


Figure 2: Our work consists of three components: (a) Training data is constructed using Grounding DINO and GPT-4o to generate structured alignment annotations; (b) A two-stage training pipeline performs reinforcement-guided visual reasoning via GRPO; (c) The model is evaluated on four fine-grained alignment benchmarks.

each e_i in the set $\langle e_i, a_i \rangle_{i=1}^{N}$, we utilize Grounding DINO (Liu et al., 2024), a state-of-the-art object grounding model, to associate the element with a corresponding region in the generated image $\mathcal{I}$. This grounding step produces a list of bounding boxes $\{b_{i,j} = [x_1, y_1, x_2, y_2]\}_{j=1}^{K_i}$ that spatially localize the element e_i within the image, where K_i denotes the number of detected regions associated with e_i. We employ a *strict grounding strategy* by raising the detection confidence threshold, which yields an empty list for low-confidence regions, effectively preventing error propagation caused by incorrect localization (see Sec. 4.4 for details). A detailed analysis of the bounding box annotation quality is presented in Sec. 4.4.

Subsequently, for each element e_i, GPT-4o is conditioned on the input tuple $(\mathcal{I}, \mathcal{P}, e_i, b_i)$ to generate a natural language explanation r_i and a predicted alignment label $\hat{a}_i$. If the associated bounding box set b_i is an empty list ([]), the model is explicitly prompted to perform reasoning based on the global visual context of the image $\mathcal{I}$. Otherwise, the reasoning focuses on the specific localized regions. To ensure the high quality of the generated reasoning rationales r_i, we employ a two-stage quality assurance strategy to strictly filter out low-quality samples (see Appendix A.1 for details). By following the above procedures, we finally curate a dataset comprising 25K high-quality samples, each annotated with a three-stage visual reasoning trajectory, denoted as $\mathcal{D}_{\text{VisualReason}}$.

3.2 Cold-Start Training with Automatically Constructed Visual Reasoning Trajectory.

To enable the MLLM to follow the proposed three-stage visual reasoning format, we first introduce a cold start training phase, in which the MLLM is fine-tuned on the automatically constructed visual reasoning trajectory $\mathcal{D}_{\text{VisualReason}}$.

Specifically, we sample a subset of 5,000 annotated instances from $\mathcal{D}_{\text{VisualReason}}$, comprising 2,500 real and 2,500 synthetic image–prompt pairs, denoted as $\mathcal{D}_{\text{SFT}}$. The selected samples are curated to ensure diversity across a wide range of element types, such as objects, attributes, and spatial. The cold start training uses supervised fine-tuning (SFT) to minimize the negative log-likelihood (NLL) of the token sequence. Formally, given $\mathcal{D}_{\text{SFT}}$, the model is trained on it to output a structured sequence of the form: $<\boldsymbol{box}>$[[x_1, y_1, x_2, y_2], ...]$</\boldsymbol{box}>$ $<\boldsymbol{thinking}>$reasoning process$</\boldsymbol{thinking}>$ $<\boldsymbol{score}>s \in [0,1]</\boldsymbol{score}>$, where $<\boldsymbol{box}>$ denotes the predicted bounding box, $<\boldsymbol{thinking}>$ is a free-form explanation, and $<\boldsymbol{score}>$ reflects the degree of alignment, with lower scores indicating stronger misalignment. The objective is to minimize the standard negative log-likelihood (NLL) loss of the structured reasoning sequence conditioned on the input image and prompt. The detailed mathematical formulation is provided in Appendix B.

This cold start training phase equips the model

with the ability to follow the visual reasoning format, establishing a baseline for subsequent RL-based optimization.

3.3 Visual Reasoning for Element-Level Text-Image Alignment Evaluation via Reinforcement Learning

While cold start training provides a baseline for element-level text-image alignment, its ability to incentivize deep reasoning capabilities in foundation models has been shown to be inferior to that of reinforcement learning (Ma et al., 2025; Wang et al., 2025a; Zhan et al., 2025). To further enhance the model's visual reasoning performance, we introduce an RL stage based on GRPO (Shao et al., 2024), equipped with a task-specific reward function and a challenging-sample selection strategy.

Challenging-sample Selection. To improve training quality, we retain only challenging samples for the reinforcement learning stage. Specifically, we use the cold-start model to generate alignment predictions on $\mathcal{D}_{\text{VisualReason}}$, and filter out data where the model accurately judges the alignment status of all elements. Only examples with at least one incorrectly predicted element are retained for the GRPO training. This results in a curated subset of 20K hard cases from the EvalMuse-40K dataset, denoted as $\mathcal{D}_{\text{Challenging-Sample}}$, used to optimize the model's alignment policy.

Reward Shaping. Given the rich supervision signals in $\mathcal{D}_{\text{Challenging-Sample}}$, we construct a composite reward function to guide the model's behavior along multiple dimensions:

(1) Format Reward evaluates whether the generated output adheres to the required structured format, including grounding stage (<***box***></***box***>), reasoning stage (<***thinking***></***thinking***>), and conclusion stage (<***score***></***score***>). Specifically, we assign a binary reward $r_{\text{format}} \in \{0, 1\}$, where $r_{\text{format}} = 1$ if the output format is correct and $r_{\text{format}} = 0$ otherwise.

(2) Box Reward quantifies the localization accuracy of predicted bounding boxes by comparing them with ground-truth annotations. It adopts a commonly used matching-based strategy to compute the Intersection over Union (IoU) between predicted and ground-truth boxes. Specifically, let $\hat{B}$ and B be the predicted and ground-truth bounding box sets for each element. We first compute the pairwise IoU matrix $\mathbf{M}$, then apply the Hungarian Algorithm to find the optimal one-to-one match as follows:

$$r_{\text{box}} = \frac{1}{\max(|\hat{B}|, |B|)} \sum_{(i,j)\in\mathcal{H}} \text{IoU}(\hat{b}_i, b_j) \quad (1)$$

where $\mathcal{H}$ is the set of matched pairs returned by the Hungarian algorithm, and unmatched elements are assigned zero IoU.

(3) Element Reward evaluates the fine-grained accuracy of the predicted alignment scores. Instead of using a simple absolute difference, we adopt a squared-error based formulation to impose heavier penalties on large deviations. Specifically, for each element, the predicted scalar score $\hat{s}_i$ is compared against the reference score s_i as follows:

$$r_{\text{element}} = 1 - \frac{1}{N}\sum_{i=1}^{N}(\hat{s}_i - s_i)^2 \quad (2)$$

This formulation yields a continuous reward in the range $[0, 1]$. By utilizing the squared term, the reward provides sharper gradients for significant errors, encouraging the model to converge more strictly toward the ground truth compared to linear feedback.

By combining the aforementioned rewards, the total reward for RL training is defined as:

$$r(\tau) = \lambda_1 r_{\text{format}} + \lambda_2 r_{\text{box}} + \lambda_3 r_{\text{element}} \quad (3)$$

where λ_1, λ_2, and λ_3 are weighting hyperparameters tuned via grid search (see Appendix B for details).

Reinforcement Optimization. For policy optimization, GRPO samples a group of outputs $\{o_i\}_{i=1}^{G}$ for each query q and utilizes group-based advantage normalization. The policy π_θ is updated by maximizing the following surrogate objective:

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim \mathcal{D}_{\text{Challenging-Sample}}, \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)] \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\{\min\left[\rho_t A_t, \text{clip}(\cdot)\cdot A_t\right] \\ -\beta\mathbb{D}_{\text{KL}}\left[\pi_\theta||\pi_{\text{ref}}\right]\},\end{aligned} \quad (4)$$

where ρ_t denotes the policy ratio $\frac{\pi_\theta(o_t|q)}{\pi_{\theta_{\text{old}}}(o_t|q)}$, A_t represents the advantage normalized within the group, and $\mathbb{D}_{\text{KL}}$ is the unbiased KL-divergence estimator (Schulman, 2020). clip$(\cdot)$ refers to applying a clipping function to ρ_t that bounds it within $[1-\epsilon, 1+\epsilon]$, where ϵ is hyperparameter. Full mathematical derivations and implementation details are provided in Appendix B.

4 Experiments

4.1 Experimental Setup

Evaluation Benchmarks. We conduct experiments on four fine-grained benchmarks: EvalMuse-40K, RichHF, MHaluBench, and GenAI-Bench. Details are included in C.1.

Evaluation Metrics. To comprehensively assess model performance, we employ three metrics across all benchmarks: Spearman's Rank Correlation Coefficient (SRCC) and Pearson Linear Correlation Coefficient (PLCC) to measure the correlation between predicted scores and human judgments, alongside Accuracy (ACC) for binary classification evaluation.

Baselines. We compare our method with a series of strong baselines from two categories: **(1) Prompting-based Methods.** We include four representative approaches for text-to-image alignment evaluation: TIFA (Hu et al., 2023), VQ^2 (Yarom et al., 2023), VIEScore (Ku et al., 2024), and VQAScore (Lin et al., 2024). Additionally, we introduce a training-free variant of **REVEALER**, where Grounding DINO is utilized to extract object regions, which are subsequently passed to Gemini 3 Pro for reasoning and scalar alignment scoring. **(2) Training-based Methods.** We include FGA-BLIP2 (Han et al., 2024), a specialized end-to-end scoring model fine-tuned on the EvalMuse-40K training set to directly predict element-level alignment scores. Additionally, we establish strong supervised baselines using general MLLMs: Qwen3-VL-8B-Instruct, InternVL3-8B-hf, and LLaVA-v1.6-Mistral-7B-hf. These models are fully fine-tuned on $\mathcal{D}_{\text{VisualReason}}$ to generate the complete visual reasoning trajectory (grounding, reasoning, and conclusion).

Implementation Details. All models are trained using 8×NVIDIA H200 GPUs. Our framework demonstrates exceptional training efficiency with low resource consumption: the SFT stage (5 epochs) requires approximately 16 GPU hours, while the RL stage (3 epochs) consumes around 120 GPU hours.

4.2 Main results

Based on the results presented in Table 1, we make the following observations.

First, the zero-shot adaptation of **REVEALER** (combining Grounding DINO with Gemini 3 Pro) demonstrates superior performance compared to existing prompting-based baselines. It achieves comprehensive improvements over the strong TIFA (Gemini 3 Pro) baseline, with gains of **+1.6%** SRCC, **+2.2%** PLCC, and **+2.1%** ACC on EvalMuse-40K, validating the effectiveness of the structured visual reasoning format itself. Second, integrating GRPO training into **REVEALER** yields substantial improvements over Training-based Methods. Specifically, on EvalMuse-40K, our InternVL3-8B-hf and Qwen3-VL-8B-Instruct models outperform their respective SFT counterparts by **+13.0%** and **+13.7%** in SRCC, **+12.7%** and **+14.7%** in PLCC, and **+11.4%** and **+13.1%** in ACC. This indicates that RL effectively aligns the model's reasoning process with human preference beyond simple imitation learning. Third, our best-performing model, **REVEALER** (Qwen3-VL-8B-Instruct), achieves state-of-the-art performance across all metrics on all benchmarks. Compared to the strongest external proprietary baseline (TIFA with Gemini 3 Pro), our method establishes a clear margin, surpassing it by approximately **4.2%** SRCC, **6.0%** PLCC, and **4.0%** ACC on EvalMuse-40K, and by **7.2%** SRCC, **7.1%** PLCC, and **5.3%** ACC on RichHF. Finally, notably, our **REVEALER** models trained solely on the EvalMuse-40K dataset maintain stable high performance when evaluated on unseen benchmarks (RichHF, MHaluBench, and GenAI-Bench), demonstrating strong generalization capabilities beyond the training distribution.

4.3 Ablations

We conduct ablation studies to assess the contribution of each component in our framework. **Qwen3-VL-8B-Instruct** serves as the base model. "+ Cold Start" refers to supervised fine-tuning with formatted alignment data. "+ Reasoning" adds natural language explanation generation (<thinking>). "+ Grounding" introduces bounding box prediction (<box>) to ground observations before reasoning. "+ GRPO" (REVEALER) applies reinforcement learning to align the model with human preferences. The subtractive settings "w/o Visual Reasoning" and "w/o Challenging Sample" denote the removal of the grouding step and the Challenging-sample Selection strategy during GRPO training, respectively.

As shown in Table 2, performance generally improves with added components. Cold start and structured reasoning yield steady gains. GRPO brings the most significant improvement, with +13.2% and +7.9% accuracy gains on EvalMuse-40K and RichHF, respectively. Interestingly, visual

Method	Model	EvalMuse-40K			RichHF			MHaluBench			GenAI-Bench		
		srcc	plcc	acc	srcc	plcc	acc	srcc	plcc	acc	srcc	plcc	acc
Prompting-based Methods													
TIFA	Gemini 3 Pro	68.1	65.8	81.3	66.1	65.4	80.8	68.5	67.2	81.0	71.4	72.3	83.9
	Qwen3-VL-235B-A22B-Instruct	66.3	65.1	80.4	64.6	63.9	80.5	67.8	66.8	81.7	68.4	71.5	83.2
VQ^2	GPT-4o	67.9	66.4	81.7	63.9	64.8	77.9	65.8	67.4	80.7	70.2	71.4	84.1
	Qwen3-VL-235B-A22B-Instruct	68.1	66.9	80.9	64.4	63.1	80.3	67.2	65.6	81.5	70.7	71.8	83.0
VQAScore	CLIP-FlanT5-XXL	51.8	51.2	65.5	63.9	65.7	77.2	64.1	65.7	78.8	70.8	69.3	84.1
VIEScore	GPT-4o	65.3	66.5	80.2	65.8	66.2	79.1	67.8	66.2	81.7	69.2	68.6	82.9
Training-based Methods													
FGA-BLIP2	BLIP2	62.1	64.6	76.8	56.6	57.9	71.4	63.2	65.1	77.7	65.3	66.9	79.0
SFT	Qwen3-VL-8B-Instruct	58.6	57.1	72.2	63.4	63.9	76.7	65.2	66.7	79.3	67.1	65.7	80.3
	InternVL3-8B-hf	57.4	56.8	72.5	60.2	61.4	75.8	65.9	65.2	78.2	67.8	68.1	78.1
	LLaVA-v1.6-7B-hf	54.7	55.2	73.1	55.7	57.4	70.9	61.7	62.5	74.3	62.9	63.5	76.5
REVEALER *(Ours)*													
REVEALER	DINO + Gemini 3 Pro	69.7	68.0	83.4	67.5	67.7	83.3	69.8	68.6	82.2	72.7	74.0	84.5
	vs. Gemini 3 Pro	(+1.6)	(+2.2)	(+2.1)	(+1.4)	(+2.3)	(+2.5)	(+1.3)	(+1.4)	(+1.2)	(+1.4)	(+1.7)	(+0.6)
	InternVL3-2B-hf	64.7	65.3	77.8	63.5	64.2	78.6	64.7	63.4	77.9	66.8	66.3	79.4
	InternVL3-8B-hf	70.4	69.5	83.9	70.8	69.2	84.7	70.6	69.3	83.4	73.4	71.9	85.0
	vs. SFT	(+13.0)	(+12.7)	(+11.4)	(+10.6)	(+7.8)	(+8.9)	(+4.7)	(+4.1)	(+5.2)	(+5.6)	(+3.8)	(+6.9)
	Qwen3-VL-4B-Instruct	66.1	65.8	80.3	68.9	67.2	81.4	66.7	67.4	80.8	69.6	68.9	82.6
	Qwen3-VL-8B-Instruct	**72.3**	**71.8**	**85.3**	**73.3**	**72.5**	**86.1**	**72.7**	**71.4**	**85.6**	**74.9**	**75.6**	**86.8**
	vs. SFT	(+13.7)	(+14.7)	(+13.1)	(+9.9)	(+8.6)	(+9.4)	(+7.5)	(+4.7)	(+6.3)	(+7.8)	(+9.9)	(+6.5)

Table 1: Main results on element-level text-to-image alignment evaluation. We compare REVEALER against representative Prompting-based and Training-based baselines. **Bold** and underlined denote the best and second-best results, respectively. The rows labeled *vs.Gemini 3 Pro/SFT* highlight the absolute performance gains achieved by our method, demonstrating consistent and statistically significant improvements ($p = 0.016 < 0.05$) over standard paradigms.

Model	EvalMuse-40K		RichHF	
	SRCC	ACC	SRCC	ACC
Qwen3-VL-8B-Instruct	55.7	65.9	56.2	70.7
+ Cold Start	58.1	71.2	62.6	75.8
+ Reasoning	60.4	77.9	70.8	80.7
+ Grouding	59.8	72.1	67.5	78.2
+ GRPO (REVEALER)	**72.3**	**85.3**	**73.3**	**86.1**
REVEALER	**72.3**	**85.3**	**73.3**	**86.1**
w/o Visual Reasoning	70.1	80.1	71.2	79.6
w/o Challenging Sample	71.5	82.0	72.8	84.1

Table 2: Ablation studies on **EvalMuse-40K** and **RichHF** benchmarks (SRCC% and Acc%).

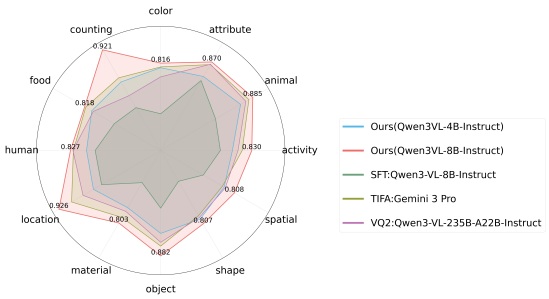


Figure 3: Accuracy across different element categories on the EvalMuse-40K benchmark.

reasoning without GRPO hurts performance, likely due to incorrect visual grounding leading to flawed reasoning. This is validated by the "w/o Visual Reasoning" setting, which results in drops of 5.2% and 6.5% on the two benchmarks. Finally, disabling challenging sampling leads to performance drops of 3.3% and 2.0%, confirming its positive effect on training quality.

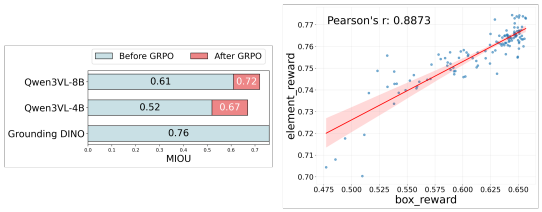
4.4 Analyses

Performance Across Different Element Categories. We evaluate alignment performance across different categories in EvalMuse-40K. As shown in Figure 3, our model (Qwen3-VL-8B-instruct) achieves superior performance, particularly in concrete categories like *counting* and *location*, validating the effectiveness of the structured **grounding–reasoning–conclusion** paradigm.

Effect of Strict Grounding Strategy. To further address the challenge of localizing abstract concepts, we propose a *Strict Grounding Strategy* by elevating the confidence threshold of Grounding DINO ($\gamma = 0.35 \rightarrow 0.55$). This encourages the output of empty box lists (`[]`) when visual evidence is ambiguous, which prevents grounding error propagation and encourages the model to

Method	Threshold (γ)	Empty Box Rate (%) Group A (Concrete)	Empty Box Rate (%) Group B (Abstract)	Alignment Accuracy (%) Group A (Concrete)	Alignment Accuracy (%) Group B (Abstract)
Baseline (Forced Grounding)	0.35	2.1	12.4	86.3	80.5
REVEALER (strict Grounding)	**0.55**	**4.5**	**53.0**	**87.2**	**84.7**
Δ	-	+2.4	+40.6	+0.9	+4.2

Table 3: Impact of Strict Grounding Strategy ($\gamma = 0.35 \rightarrow 0.55$). Group A and B denote concrete and abstract elements, respectively.



(a) (b)

Figure 4: (a) Visual grounding capability before and after GRPO training. (b) Pearson correlation between box and element rewards.

switch to global reasoning for abstract concepts. As detailed in Table 3, this strategy increases the Empty Box Rate for abstract elements (Group B) from 12.4% to 53.0% while preserving precision for concrete ones (Group A). This strict Grounding mechanism yields a substantial **+4.2%** accuracy gain on abstract attributes.

Quality Validation of Visual Grounding Annotations. To validate the reliability of the bounding box annotations used in our automated data curation pipeline, and to provide a benchmark for evaluating visual grounding improvements, we constructed a high-quality, box-annotated evaluation set. Specifically, we constructed the evaluation set, denoted as $\mathcal{D}_{\text{BoxEval}}$, by randomly sampling a total of 2,000 image-prompt pairs from the EvalMuse-40K and RichHF benchmarks. We manually annotated the bounding boxes for the target elements within these pairs; notably, for abstract elements or elements absent from the image, we explicitly annotated the bounding box list as empty. To ensure precision, all annotations underwent a secondary review and correction process. As illustrated in Figure 4(a), we evaluated the performance of Grounding DINO on this human-verified set. The model achieved a mIoU of 0.76, indicating a high degree of overlap with human annotations. This result confirms the reliability of using Grounding DINO for large-scale training data synthesis.

Visual Grounding Capability Before and After Training. We evaluate the visual grounding performance of our models on $\mathcal{D}_{\text{BoxEval}}$. As shown in Figure 4(a), we find that GRPO training significantly enhances localization capabilities, boosting mIoU by **+0.11** and **+0.15** for the 4B and 8B models, respectively. Furthermore, we analyze the relationship between grounding precision and evaluation accuracy. The results reveal a strong positive correlation (Pearson $r = 0.8773$) between grounding accuracy and alignment scores (Figure 4(b)), confirming that precise visual reasoning directly contributes to more accurate alignment evaluation.

Grounding Status	Condition *(Filter Criteria)*	Distribution	Reasoning Hal. Rate↓	Alignment Acc↑
Accurate Grounding	MIoU ≥ 0.5	80.5%	**8.4%**	**89.6%**
Misleading Grounding	MIoU < 0.5	8.1%	46.2%	76.3%
Strict Grounding Strategy	Empty Box ([])	12.4%	14.7%	81.2%

Table 4: Visual Grounding Error Propagation Analysis (Qwen3-VL-8B).

Visual Grounding Error Propagation Analysis. To investigate error propagation from visual grounding to downstream reasoning, we conducted a detailed analysis using $\mathcal{D}_{\text{BoxEval}}$. Specifically, to ensure metric reliability, we manually verified the reasoning traces to identify hallucinations. As detailed in Table 4, *Misleading Grounding* (MIoU $<$ 0.5) triggers severe error propagation, spiking the reasoning hallucination rate to **46.2%** and drastically dropping alignment accuracy to **76.3%**. This confirms that incorrect visual cues actively mislead the reasoning process. In contrast, our **Strict Grounding Strategy** acts as a safety mechanism by suppressing low-confidence predictions, effectively shifting high-risk samples to *Global Reasoning*. This fallback mechanism significantly reduces hallucinations to **14.7%** and recovers alignment accuracy to **81.2%**, demonstrating that relying on global context is far superior to reasoning based on erroneous visual evidence.

5 Conclusion

We introduced **REVEALER**, a reinforcement-guided visual reasoning framework for element-level text-to-image alignment evaluation. By enforcing a structured "grounding–reasoning–conclusion" paradigm and optimizing via GRPO, our approach effectively bridges the gap between visual localization and semantic judgment. Experiments across four benchmarks show that REVEALER achieves state-of-the-art performance, surpassing proprietary models like Gemini 3 Pro.

Limitations

Despite the superior performance of REVEALER, several limitations remain. First, the explicit box-based grounding paradigm is optimized for concrete semantic elements and may be less naturally suited for evaluating holistic qualities, such as artistic style, complex lighting atmospheres, or emotional tone, where discrete localization is ambiguous. Furthermore, our current work focuses exclusively on static image-text alignment; consequently, the applicability of our framework to text-to-video alignment evaluation is limited, as it does not account for temporal dynamics or motion consistency. Future work will aim to extend the visual reasoning framework to address these non-localized and temporal challenges.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. 2025. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *Preprint*, arXiv:2505.14231.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Unified hallucination detection for multimodal large language models. *Preprint*, arXiv:2402.03190.

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

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. 2024. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. *Preprint*, arXiv:2310.18235.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. Scaling rectified flow transformers for high-resolution image synthesis. *Preprint*, arXiv:2403.03206.

Shuhao Han, Haotian Fan, Jiachen Fu, Liang Li, Tao Li, Junhui Cui, Yunqiu Wang, Yang Tai, Jingwei Sun, Chunle Guo, and Chongyi Li. 2024. Evalmuse-40k: A reliable and fine-grained benchmark with comprehensive human annotations for text-to-image generation model evaluation. *Preprint*, arXiv:2412.18150.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2022. Clipscore: A reference-free evaluation metric for image captioning. *Preprint*, arXiv:2104.08718.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Preprint*, arXiv:1706.08500.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. 2023. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. *Preprint*, arXiv:2303.11897.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. 2025. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *Preprint*, arXiv:2307.06350.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. 2023. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Preprint*, arXiv:2305.01569.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. 2024. Viescore: Towards explainable metrics for conditional image synthesis evaluation. *Preprint*, arXiv:2312.14867.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. 2024. Genai-bench: Evaluating and improving compositional text-to-visual generation. *Preprint*, arXiv:2406.13743.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

Weiqi Li, Xuanyu Zhang, Shijie Zhao, Yabin Zhang, Junlin Li, Li Zhang, and Jian Zhang. 2025. Q-insight: Understanding image quality via visual reinforcement learning. *Preprint*, arXiv:2503.22679.

Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, Junjie Ke, Krishnamurthy Dj Dvijotham, Katie Collins, Yiwen

Luo, Yang Li, Kai J Kohlhoff, Deepak Ramachandran, and Vidhya Navalpakkam. 2024. Rich human feedback for text-to-image generation. *Preprint*, arXiv:2312.10240.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. 2024. Evaluating text-to-visual generation with image-to-text generation. *Preprint*, arXiv:2404.01291.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *Preprint*, arXiv:2303.05499.

Yujie Lu, Xianjun Yang, Xiujun Li, Xin Eric Wang, and William Yang Wang. 2023. Llmscore: Unveiling the power of large language models in text-to-image synthesis evaluation. *Preprint*, arXiv:2305.11116.

Zi-Ao Ma, Tian Lan, Rong-Cheng Tu, Shu-Hang Liu, Heyan Huang, Zhijing Wu, Chen Xu, and Xian-Ling Mao. 2025. T2i-eval-r1: Reinforcement learning-driven reasoning for interpretable text-to-image evaluation. *Preprint*, arXiv:2505.17897.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *Preprint*, arXiv:2102.12092.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. *Preprint*, arXiv:2112.10752.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Preprint*, arXiv:2205.11487.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Preprint*, arXiv:1606.03498.

John Schulman. 2020. Approximating kl divergence.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and Yu Cheng. 2025. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *Preprint*, arXiv:2505.08617.

Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, Mengping Yang, Cheng Zhang, and Hao Li. 2024. Evalalign: Supervised fine-tuning multimodal llms with human-aligned data for evaluating text-to-image models. *Preprint*, arXiv:2406.16562.

the OpenAI Team. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Rong-Cheng Tu, Zi-Ao Ma, Tian Lan, Yuehao Zhao, Heyan Huang, and Xian-Ling Mao. 2024. Automatic evaluation for text-to-image generation: Task-decomposed framework, distilled training, and meta-evaluation benchmark. *Preprint*, arXiv:2411.15488.

Yibin Wang, Zhimin Li, Yuhang Zang, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. 2025a. Unified multimodal chain-of-thought reward model through reinforcement fine-tuning. *Preprint*, arXiv:2505.03318.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. 2025b. Unified reward model for multimodal understanding and generation. *Preprint*, arXiv:2503.05236.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. 2025. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *Preprint*, arXiv:2506.09965.

Michal Yarom, Yonatan Bitton, Soravit Changpinyo, Roee Aharoni, Jonathan Herzig, Oran Lang, Eran Ofek, and Idan Szpektor. 2023. What you see is what you read? improving text-image alignment evaluation. *Preprint*, arXiv:2305.10400.

Yufei Zhan, Yousong Zhu, Shurong Zheng, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao Wang. 2025. Vision-r1: Evolving human-free alignment in large vision-language models via vision-guided reinforcement learning. *Preprint*, arXiv:2503.18013.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. 2025. Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl. *Preprint*, arXiv:2505.15436.

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

A Dataset Details

A.1 Quality Assurance for Visual Reasoning Trajectory

To ensure the high quality of the generated reasoning rationales r_i, we employ a two-stage quality assurance strategy to strictly filter out low-quality samples. First, in the *self-correction stage*, if the predicted label $\hat{a}_i$ is inconsistent with the ground-truth label a_i, we re-prompt the model to generate a revised explanation and prediction. Data points that fail to reach consistency after three attempts are discarded, and for the retained samples, we adopt the human-annotated label a_i from EvalMuse-40K as the final ground truth. Second, in the *logical verification stage*, we employ Gemini 3 pro to further guarantee logical coherence by verifying the consistency between the generated r_i and the label a_i. Specifically, the model assesses whether r_i logically supports a_i, and any data points exhibiting logical inconsistencies are strictly filtered out.

B Training Details

Hyperparameter Sensitivity and Configuration To balance the multi-objective nature of our reward function, we conducted a grid search to determine the optimal scalar coefficients λ_1, λ_2, and λ_3. We observed that the model quickly learns to adhere to the structural format; therefore, we fixed the format reward weight at a low value of $\lambda_1 = 0.1$ to prevent it from dominating the optimization landscape. We then performed a grid search for the visual grounding weight (λ_2) and element alignment weight (λ_3) over the range $\{0.4, 0.45, 0.5, 0.55\}$. We evaluated the model's performance on a hold-out validation set from EvalMuse-40K. As illustrated in Figure 5, the results indicate a performance peak where slightly higher emphasis is placed on the final element alignment score. The optimal configuration was identified as $\lambda_1 = 0.1$, $\lambda_2 = 0.45$, and $\lambda_3 = 0.55$. This setting ensures that while visual grounding provides necessary evidence, the ultimate fidelity of the alignment judgment remains the primary optimization target.

SFT Objective. In the cold-start stage, we fine-tune the model to generate the structured reasoning trajectory. Let q denote the concatenation of the input inputs $(\mathcal{I}, \mathcal{P}, \{e_i\}_{i=1}^N)$, and g denote the target output sequence formed by concatenating $\{\langle b_i, r_i, a_i\rangle\}_{i=1}^N$. The training objective is to minimize the negative log-likelihood:

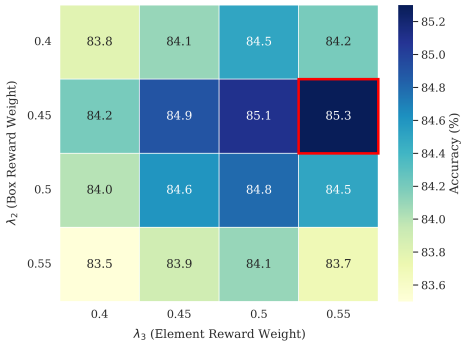


Figure 5: Grid search results for reward weights λ_2 and λ_3 with fixed $\lambda_1 = 0.1$. The heatmap shows validation accuracy on EvalMuse-40K. The red box indicates the optimal configuration ($\lambda_2 = 0.45, \lambda_3 = 0.55$).

$$\mathcal{L}_{\text{cold}} = -\mathbb{E}_{q\sim\mathcal{D}_{SFT}} \sum_{t=1}^{T} \log P_\theta(g_t \mid g_{<t}, q) \quad (5)$$

where g_t is the t-th token in the output sequence and θ denotes the model parameters.

GRPO Optimization. Given the defined total reward $r(\tau)$, we optimize the policy model using GRPO, a lightweight and stable variant of Proximal Policy Optimization (PPO). Specifically, for each $(\mathcal{I}, \mathcal{P}, \{\langle e_i, b_i, r_i, a_i\rangle\}_{i=1}^N)$ in $\mathcal{D}_{\text{Challenging-Sample}}$, a reasoning trajectory sequence τ generated by the policy model, the rule-based reward function $r(\cdot)$ computes its reward as $r(\tau)$. GRPO normalizes this scalar into an advantage $A_t = \frac{r(\tau)-\mu}{\sigma}$ for each decoding step $t \in \{1, \ldots, T\}$, where μ and σ are the batch-wise mean and standard deviation of rewards. GRPO samples a group of generated output set $\{o_1, o_2, \cdots, o_G\}$ for each q from the policy model $\pi_{\theta_{\text{old}}}$ and let the policy ratio at step t be $\rho_t = \frac{\pi_\theta(o_t|o_{i,<t},q)}{\pi_{\theta_{\text{old}}}(o_t|o_{i,<t},q)}$, where o_i represents the outputs sampled from the policy model. The trained policy π_θ is then updated by maximizing the following objective:

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim \mathcal{D}_{\text{Challenging-Sample}}, \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)] \\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\{\min\left[\rho_t A_t, \text{clip}(\cdot)\cdot A_t\right] \\ -\beta\mathbb{D}_{\text{KL}}\left[\pi_\theta||\pi_{\text{ref}}\right]\},\end{aligned} \quad (6)$$

Here, π_{ref} denotes the frozen reference policy obtained from the SFT stage. clip$(\cdot)$ refers to applying a clipping function to ρ_t that bounds it within

$[1-\epsilon, 1+\epsilon]$, where ϵ is hyperparameter. This clip function helps prevent excessively large policy updates. Unlike the KL penalty term used in (Ouyang et al., 2022), we estimate the KL divergence with the unbiased estimator (Schulman, 2020), which is guaranteed to be positive. We set $\epsilon = 0.2$ and $\beta = 1e-2$ during training. The hyperparameter β controls the KL divergence penalty, which encourages the new policy to stay close to the reference policy, thereby stabilizing training.

C Evaluation Details

C.1 Benchmark Details

We evaluate alignment accuracy across four fine-grained benchmarks: EvalMuse-40K, RichHF, MHaluBench, and GenAI-Bench. (1) **EvalMuse-40K** provides element-level alignment annotations across categories such as object, attribute, and location. Each element is labeled as aligned (1) or unaligned (0) by multiple annotators, and final labels are averaged; elements with scores ≥ 0.5 are considered aligned. (2) **RichHF** (Liang et al., 2024) offers keyword-level annotations over diverse prompt styles. We evaluate on the annotated subset using accuracy. (3) **MHaluBench** (Chen et al., 2024) provides claim-level annotations. To enable fine-grained evaluation, we extract elements via GPT-4 (the OpenAI Team, 2024), generate binary questions, and collect human annotations following the EvalMuse-40K protocol. (4) **GenAI-Bench** (Li et al., 2024) targets complex compositional prompts. As it lacks element-level labels, we apply the same procedure as in MHaluBench.

C.2 Adaptation of Benchmarks for Fine-Grained Evaluation

To support element-level multimodal hallucination detection, we adapted two existing benchmarks—**MHaluBench** and **GenAI-Bench**—by applying a unified annotation protocol inspired by EvalMuse-40K (Han et al., 2024). While MHaluBench (specifically its text-to-image subset) and GenAI-Bench provide diverse prompting schemes, they originally lack granular semantic annotations. To address this, we decompose each natural language prompt into discrete semantic elements using GPT-4, categorizing them according to the TIFA taxonomy (e.g., object, attribute, spatial). For each element, we generate a corresponding binary verification question (e.g., "Is there a red car in the image?") to assess visual fidelity. These element-question pairs undergo rigorous human verification to determine semantic alignment (labeled as 1 for aligned, 0 for misaligned), thereby enabling consistent, interpretable, and fine-grained evaluation across both compositional and general scenarios.

Method	RichHF		MHaluBench		GenAI-Bench	
	srcc↑	time↓	srcc↑	time↓	srcc↑	time↓
Chain-of-Focus	65.1	5.9	67.3	7.3	69.1	6.6
ViLaSR	64.2	6.5	65.4	5.7	68.9	6.2
Vision-R1	64.7	5.9	66.2	6.8	68.0	4.8
Q-Insight	<u>67.4</u>	<u>4.5</u>	<u>67.9</u>	<u>4.1</u>	<u>70.3</u>	<u>4.4</u>
REVEALER	**70.8**	**1.3**	**70.6**	**1.6**	**74.4**	**1.2**

Table 5: Comparison with RL-based visual reasoning methods. **Time** denotes the average inference latency per sample measured on a single A800 GPU.

C.3 Adaptation of Zero-Shot Baselines for Element-Level Evaluation

To ensure a rigorous comparison, we adapt representative zero-shot methods—TIFA, VQ^2, VQAScore, and VIEScore—to our fine-grained evaluation task through a unified pipeline. For each baseline, we first employ a large language model (GPT-4) to decompose the input prompt into discrete, visually verifiable semantic units according to the TIFA taxonomy, such as objects, attributes, and spatial. These units are subsequently converted into method-specific query formats, ranging from binary VQA questions to structured semantic triples. Finally, pre-trained multimodal models are utilized to verify the visual grounding of each query against the generated image. This process standardizes the output into binary alignment labels for individual semantic elements, facilitating a consistent and interpretable performance assessment across all methods.

D Additional Analyses

D.1 Comparison with RL-based Visual Reasoning Methods.

We compare REVEALER against representative RL-based MLLMs, including Chain-of-Focus (Zhang et al., 2025), ViLaSR (Wu et al., 2025), Vision-R1 (Zhan et al., 2025), and Q-Insight (Li et al., 2025). As shown in Table 5, our method establishes a superior trade-off between alignment accuracy and computational efficiency. **Accuracy.** Existing iterative methods (Chain-of-Focus, ViLaSR) suffer from error propagation during multi-turn in-

teractions, while Q-Insight focuses more on the global image quality score. By anchoring reasoning to specific elements via a structured paradigm, REVEALER mitigates these issues, surpassing the strongest baseline (Q-Insight) by significant margins of **+3.4%** and **+4.1%** SRCC on RichHF and GenAI-Bench, respectively. **Efficiency.** Unlike baselines that require multiple forward passes for visual resampling or unconstrained reasoning generation, REVEALER integrates localization and reasoning into a single cohesive pass. This streamlined architecture reduces inference time to **1.2s–1.6s** per sample, representing a significant efficiency gain over preceding RL-based methods.

D.2 Impact of Continuous vs. Binary Rewards on GRPO Training.

To investigate the impact of reward formulation on GRPO training, we compare two designs of element-level reward for Qwen2.5-VL-3B-Instruct. The first is a binary reward, where each element is assigned 1 if the alignment prediction is correct and 0 otherwise. The second is a continuous reward, calculated as the absolute difference between the model's predicted alignment score, bounded within $[0, 1]$, and the corresponding ground-truth label. As shown in Figure 6, training with continuous rewards yields a more stable optimization process and outperforms binary rewards by 2.2% in accuracy on the EvalMuse-40K benchmark. We attribute this improvement to the finer-grained feedback provided by continuous rewards. In particular, continuous rewards lead to smoother reward landscapes and more reliable policy updates, especially during early training when binary signals are often sparse or uninformative.

D.3 Statistical Significance Analysis

To rigorously validate that the performance gains of our proposed method over strong baselines (DINO, Gemini 3 Pro) stem from the effective reinforcement-guided reasoning framework rather than random variance, we conducted a statistical significance test using a stratified bucketing approach. Specifically, we randomly partitioned the EvalMuse-40K test set into $K = 10$ disjoint folds and computed the element-level alignment accuracy for both our GRPO-optimized model (Qwen3-VL-8B-Instruct) and the Gemini 3 Pro baseline on each fold. We then performed a one-sided Wilcoxon Signed-Rank Test on the resulting paired accuracy distributions to assess the consistency of

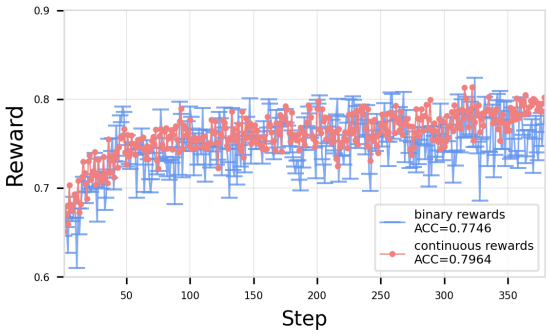


Figure 6: Comparison of training stability and final accuracy between binary and continuous reward designs. Continuous rewards result in more stable training and better alignment performance on EvalMuse-40K, achieving an ACC of 0.7964 compared to 0.7746 with binary rewards.

the improvement. The analysis yielded a p-value of **0.016** ($p < 0.05$). This result statistically rejects the null hypothesis, confirming that our method's improvement is statistically significant and robust across different data distributions.

E Prompt Templates

Prompt for Visual Reasoning Trajectory Generation

```
System Instruction:
You are an expert evaluator for text-to-image alignment. Your task is to perform visual reasoning
    to determine if a specific element (e_i) from the input prompt (P) is accurately represented
    in the generated image (I). You are provided with bounding boxes (b_i) detected by a grounding
    model.

Input Data:
- Full Prompt (P): {full_prompt}
- Target Element (e_i): {element}
- Bounding Boxes (b_i): {box_list} (Format: [[x1, y1, x2, y2]...])

Reasoning Rules:
1. Localized Reasoning (If b_i is NOT empty):
   - Focus strict attention on the visual content within the provided coordinates.
   - Verify if the visual element inside the boxes match the description of e_i.
   - Ignore background details outside the boxes unless they directly affect the element's state.

2. Global Reasoning (If b_i is empty []):
   - Switch to Global Context Analysis. The grounding model failed to localize the element.
   - Scenario A (Concrete Object): If e_i is a tangible object (e.g., "cat", "car"), its absence
     usually implies misalignment. Verify if it is truly missing.
   - Scenario B (Abstract Attribute/Style): If e_i is global (e.g., "foggy", "oil painting", "
     lighting"), evaluate the entire image atmosphere. Empty boxes are expected here.

Output Format:
Return a JSON object containing:
- "reasoning" (r_i): A step-by-step rationale based on the rules above.
- "label" (â_i): 1 for Aligned, 0 for Misaligned.

Response:
```

Figure 7: The system prompt template used for visual reasoning trajectory curation. The prompt explicitly instructs the model to handle both grounded (localized) and ungrounded (global) scenarios.

Prompt for Visual Reasoning Self-Correction

```
System Instruction:
You are an expert evaluator for text-to-image alignment. You are provided with a Reference
    Alignment Label (a_i) derived from human annotation for a specific element (e_i). Your task is
    to re-examine the image and bounding boxes (b_i) to construct a visual reasoning path that
    logically supports this reference label.

Input Data:
- Full Prompt (P): {full_prompt}
- Target Element (e_i): {element}
- Bounding Boxes (b_i): {box_list}
- Reference Label (a_i): {ground_truth_label} (1 = Aligned, 0 = Misaligned)

Reasoning Rules:
1. Localized Reasoning (If b_i is NOT empty):
   - Focus strict attention on the visual content within the provided coordinates.
   - Verify if the visual element inside the boxes match the description of e_i.
   - Ignore background details outside the boxes unless they directly affect the element's state.

2. Global Reasoning (If b_i is empty []):
   - Switch to Global Context Analysis. The grounding model failed to localize the element.
   - Scenario A (Concrete Object): If e_i is a tangible object (e.g., "cat", "car"), its absence
     usually implies misalignment. Verify if it is truly missing.
   - Scenario B (Abstract Attribute/Style): If e_i is global (e.g., "foggy", "oil painting", "
     lighting"), evaluate the entire image atmosphere. Empty boxes are expected here.

Correction Rules:
1. Evidence Re-Discovery:
   - If Reference (a_i) is 1 (Aligned): Look closely at the region/image to identify the specific
     visual features (color, shape, count) that confirm the element's presence.
   - If Reference (a_i) is 0 (Misaligned): Look for the specific visual discrepancy (e.g., wrong
     color, missing object, distorted shape) that contradicts the prompt.

2. Strict Formatting Constraint (Crucial):
   - Your reasoning must be self-contained and based solely on visual observation.
   - DO NOT mention the "Reference Label," "Human Annotation," or "Ground Truth" in your reasoning
     text.
   - DO NOT write phrases like "As indicated by the reference..." or "Since the label is 1..."
   - Simply state the visual facts that lead to the conclusion.

Output Format:
Return a JSON object containing:
- "reasoning" (r_i): A factual visual analysis describing *why* the image conforms to the Reference
    Label.
- "label" (â_i): The final label (should match a_i).

Response:
```

Figure 8: The self-correction prompt template. When the initial prediction disagrees with the ground truth, the model is guided to re-evaluate the visual evidence to align with the human annotation (a_i) without explicitly referencing the hint in the rationale.

Prompt for Logical Consistency Verification

```
System Instruction:
You are a Quality Assurance Auditor for an automated evaluation system. Your task is to verify the
     logical consistency between a generated reasoning rationale (r_i) and its assigned binary
    label (a_i) for a target element (e_i). You must detect contradictions between the textual
    explanation and the numerical score.

Input Data:
- Target Element (e_i): {element}
- Generated Reasoning (r_i): {reasoning_text}
- Assigned Label (a_i): {label} (1 = Aligned, 0 = Misaligned)

Verification Rules:
1. Logical Entailment Check:
   - Does the text in r_i explicitly state that the element is correctly depicted or aligned? If yes
     , a_i must be 1.
   - Does the text in r_i describe missing objects, wrong attributes, or hallucinations? If yes, a_i
     must be 0.

2. Identify Contradictions:
   - Flag as "Inconsistent" if r_i describes a failure (e.g., "The car is blue instead of red") but
     a_i is 1.
   - Flag as "Inconsistent" if r_i describes a success (e.g., "The car is correctly rendered in red
     ") but a_i is 0.

Output Format:
Return a JSON object containing:
- "is_consistent": boolean (true/false)
- "analysis": "Brief explanation of the consistency check."

Response:
```

Figure 9: The logical verification prompt used by Gemini 3 Pro. This step filters out low-quality samples where the generated reasoning text (r_i) logically contradicts the final classification label (a_i).

Prompt for End-to-End Element Alignment Inference

```
<image>
System Instruction:
You are an expert in fine-grained text-to-image alignment evaluation. Your task is to perform
    Element-level Hallucination Detection on the provided image based on the input prompt.

Input Data:
- Prompt (𝒫): {original_prompt}
- Target Elements (E): {element_keys_str}

Evaluation Protocol:
For each element in the target list, perform the following steps sequentially:
1. Localization (<box>):
   - Identify the element's location in the image.
   - Output bounding boxes in the format [[x1, y1, x2, y2]...] .
   - If the element is missing or abstract (unable to be grounded), output an empty list [].

2. Visual Reasoning (<thinking>):
   - Analyze whether the visual depiction matches the textual description (appearance, action,
     relation).
   - Explicitly state any discrepancies (e.g., "present but wrong color", "missing entirely").

3. Scoring (<score>):
   - Assign a fidelity score between 0.0 and 1.0.
   - 1.0 = Perfectly present and accurate.
   - 0.0 = Entirely missing or hallucinated.

Output Format:
Output a single Python dictionary string wrapped in <element> tags.
- Keys: Element names (Categories).
- Values: A concatenated string containing the tags <box>...</box><thinking>...</thinking><score
    >...</score>.

Example Output:
<element>
{
  "Eating (activity)": "<box>[[221, 162, 893, 675]]</box><thinking>The subject has food but is not
     performing the action of eating.</thinking><score>0.4</score>",
  "Puffin (animal)": "<box>[[1, 10, 486, 365]]</box><thinking>The puffin is rendered clearly but
    is in the wrong spatial location.</thinking><score>0.3</score>",
  "Pink tree (object)": "<box>[[122, 95, 900, 883]]</box><thinking>The tree matches the color and
    style description perfectly.</thinking><score>1.0</score>"
}
</element>

Constraint:
Do not include any conversational text outside the <element> tags. Ensure the JSON syntax is valid.


Response:
```

Figure 10: The inference prompt used for evaluating text-to-image models. It enforces a strict "Grounding-Reasoning-Scoring" format output within a structured dictionary for automated parsing.