ENDOGENOUS REPROMPTING: SELF-EVOLVING COGNITIVE ALIGNMENT FOR UNIFIED MULTIMODAL MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified Multimodal Models (UMMs) exhibit strong understanding, yet this capability often fails to effectively guide generation. We identify this as a Cognitive Gap: the model lacks the understanding of how to enhance its own generation process. To bridge this gap, we propose Endogenous Reprompting, a mechanism that transforms the model's understanding from a passive encoding process into an explicit generative reasoning step by generating self-aligned descriptors during generation. To achieve this, we introduce SEER (Self-Evolving Evaluator and Reprompter), a training framework that establishes a two-stage endogenous loop using only 300 samples from a compact proxy task, Visual Instruction Elaboration. First, Reinforcement Learning with Verifiable Rewards (RLVR) activates the model's latent evaluation ability via curriculum learning, producing a reliable endogenous reward signal. Second, Reinforcement Learning with Model-rewarded Thinking (RLMT) leverages this signal to optimize the generative reasoning policy. Experiments show that SEER consistently outperforms state-of-the-art baselines in evaluation accuracy, reprompting efficiency, and generation quality, without sacrificing general multimodal capabilities.

1 INTRODUCTION

Recent Unified Multimodal Models (UMMs) have successfully integrated understanding and generation capabilities into a single framework (Xie et al., 2024; Dong et al., 2023). By sharing a common backbone, UMMs possess the potential for seamless cross-modal interaction, offering a promising path to resolving the semantic disconnect often found in traditional separate-encoder T2I models. However, a significant imbalance persists: despite a shared architecture, the model's ability to generate lags substantially behind its ability to understand (Chen et al., 2025b; Pan et al., 2025; Xie et al., 2025b; Wang et al., 2024a; Jin et al., 2025). As illustrated in Figure 1, while the model accurately comprehends visual instructions, it struggles to translate this understanding into generative guidance, resulting in misalignment in the final output. We identify this as a cognitive gap: the model lacks the specific understanding of how to enhance its own generation process (details in the Appendix A.8.3). Recent studies have explored self-evolving mechanisms in UMMs, where the model leverages its own understanding capability to improve generation through internal feedback or supervision (Han et al., 2026; Fang et al., 2026). These works demonstrate the potential of endogenous optimization in unified models. While these methods focus on improving generation itself, SEER focuses on the intermediate understanding-to-generation process, using internal evaluation to optimize model-aligned reprompting.

To bridge this gap, we propose Endogenous Reprompting, a mechanism that transforms the model's understanding from a passive encoding process into an explicit generative reasoning step by translating user intention into self-aligned descriptors (aligned with the model's own capabilities) during generation. Prior approaches (Betker et al., 2023; Wang et al., 2025a) rely on disjoint models (e.g., an LLM and a generator), which leads to representation mismatch, causing the generated descriptions to mismatch the generator's priors in Figure 1. In contrast, our approach exploits the intrinsic representational consistency of UMMs. Being endogenous, all reasoning operates within the model's shared representation space, enabling knowledge to be seamlessly transferred between understanding and generation to generate reprompts. This promotes Model-Specific Alignment: the model translates

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

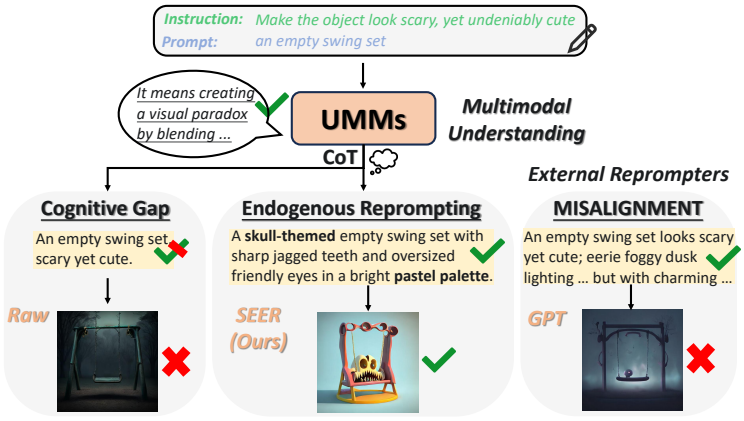


Figure 1: Bridging the cognitive gap. While direct generation fails to reflect the model's understanding (Left), and external reprompters often cause misalignment by generating descriptors that mismatch the generator's priors (Right), SEER (Middle) leverages Endogenous Reprompting. It produces concrete, self-aligned descriptors that match the generator's generative priors, successfully bridging understanding and generation.

user intention into specific linguistic descriptors that are better aligned with the generator's generative priors. We argue that this constitutes the missing "understanding-to-generation" pathway.

To achieve this, we propose SEER (Self-Evolving Evaluator and Reprompter), a training framework that establishes a two-stage endogenous loop using only 300 samples from a compact proxy task, Visual Instruction Elaboration. We introduce this task to specifically bootstrap generative reasoning: it targets the explicit elaboration of implicit user intention, forcing the model to reason about visual instructions to enhance generation rather than simply mapping text to pixels. Using this compact signal, SEER unfolds in two stages. First, we employ Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024; Wen et al., 2025) through a curriculum learning strategy to activate the internal evaluator. By adopting a pairwise optimization strategy that learns from relative comparisons, we transform the model's understanding into a reliable reward signal. Second, we employ Reinforcement Learning with Model-rewarded Thinking (RLMT) (Bhaskar et al., 2025) to harness this signal, enabling the model to autonomously refine its reasoning policy. This distinguishes our work from traditional Reinforcement learning from human feedback (RLHF) for visual generation (Ziegler et al., 2019; Xu et al., 2023; Liu et al., 2025; Gong et al., 2025): whereas RLHF targets the visual generation execution phase (optimizing pixels), our approach optimizes the generative reasoning process (optimizing prompts), effectively teaching the model to think before it generates.

Empirically, SEER enhances instruction compliance and reasoning capabilities, utilizing the model's own understanding to resolve the cognitive gap. The entire self-evolution process relies on minimal data, highlighting the efficiency of unlocking intrinsic potential rather than learning from scratch. Our key contributions are summarized as follows:

- We introduce Endogenous Reprompting, a generative reasoning framework for UMMs that bridges the cognitive gap via a compact proxy task, Visual Instruction Elaboration.
- We propose SEER, a self-evolving framework that establishes a two-stage endogenous loop to achieve Endogenous Reprompting, activating an internal evaluator with RLVR and optimizing the reasoning policy with RLMT.
- We demonstrate that SEER consistently outperforms state-of-the-art baselines in instruction compliance, reprompting efficiency, and generation quality, using only 300 samples, without sacrificing general multimodal capabilities.

2 Related Work

Unified Multimodal Models. UMMs integrate understanding and generation within a single end-to-end framework to promote cross-modal interaction. Recent research explores a broad architectural design space, primarily categorized into three paradigms: (1) Autoregressive Models, such as Janus

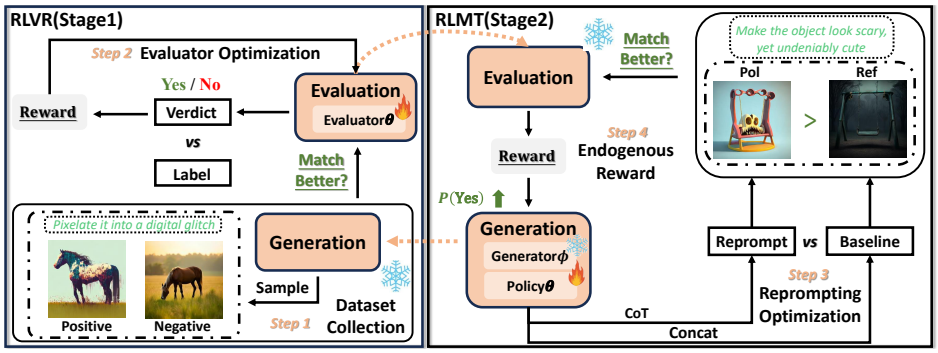


Figure 2: Overview of SEER. The framework bridges the cognitive gap via a two-stage loop. RLVR: (Step 1) collecting endogenous image pairs to (Step 2) optimize the internal evaluator. RLMT: (Step 3) optimizing the reprompting policy using (Step 4) relative rewards derived from the evaluator.

(Wu et al., 2025a), and Emu3 (Wang et al., 2024a), which tokenize inputs for sequential next-token prediction, with variants like Show-o (Xie et al., 2024); (2) Hybrid Architectures, which either attach diffusion heads to a shared transformer backbone (Zhao et al., 2024; Chen et al., 2024) or route features from frozen MLLMs to external generators via learnable queries (Tong et al., 2025; Chen et al., 2025a); and (3) Emerging Designs, including Masked-Autoregressive (MAR) generation (Li et al., 2024; Wu et al., 2025c) and scalable Mixture-of-Transformers (MoT) frameworks (Zhou et al., 2024; Deng et al., 2025). Despite these architectural strides, a common bottleneck persists: the alignment between strong understanding and generation remains insufficiently activated.

Prompt Optimization and Reprompting. To bridge the gap between user intent and generative execution, researchers have increasingly prioritized Prompt Optimization. Foundational approaches like DALL-E 3 (Betker et al., 2023) utilize recaptioning to generate dense captions for model fine-tuning. For inference-time enhancement, on-the-fly rewriting has become a dominant paradigm, employing LLMs for iterative critique and refinement (Wu et al., 2024), automated prompt engineering (Cao et al., 2023), or complex agent-based planning (Wang et al., 2024b). Recent works like PromptEnhancer (Wang et al., 2025a) further advance this by training dedicated rewriters against fine-grained reward models. However, these methods typically couple disjoint models, leading to representation mismatch between the reprompter and generator. In contrast, by leveraging the intrinsic representational consistency of UMMs, SEER activates an endogenous mechanism to generate reprompts aligned with the generative priors (i.e., model-specific alignment).

Reinforcement Learning for Visual Generation. Aligning generative models with human preferences is a critical research frontier. Early works like Imagereward (Xu et al., 2023) fine-tune diffusion models using preference scores, while Diffusion-DPO (Wallace et al., 2024) directly optimizes policies without explicit reward models. Further advances like DDPO (Black et al., 2024) and Flow-GRPO (Liu et al., 2025) apply reinforcement learning directly to the image generation process, with recent works like OneReward (Gong et al., 2025) and RewardDance (Wu et al., 2025b) leveraging VLMs to provide feedback. Crucially, existing methods primarily focus on the execution phase by optimizing diffusion trajectories. SEER diverges by applying RLMT to the generative reasoning phase. We demonstrate that evolving the generative reasoning (reprompt) rather than the low-level execution offers a more effective pathway to unlock the generative potential of UMMs.

3 METHOD

In this section, we introduce SEER (Self-Evolving Evaluator and Reprompter), a framework designed to bridge the cognitive gap in UMMs (Figure 2). Our goal is to transform the model's multimodal understanding into an active generative reasoning capability via a self-evolving loop (we also extend this loop to a second evolution round, as detailed in Appendix A.9).

3.1 PROBLEM FORMULATION

We formalize the task of Visual Instruction Elaboration as a reasoning optimization problem. Let a visual instruction be denoted by $a \in \mathcal{A}$ (e.g., "Make the object look scary, yet undeniably cute")

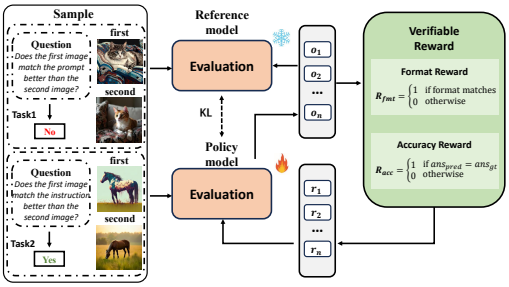


Figure 3: Stage 1: RLVR. We employ curriculum learning to transform the model into a robust internal critic. By training on pairwise comparisons (from basic alignment to instruction discrimination) using GRPO, we activate a reliable internal evaluator $E(x; a, p_0)$ capable of assessing user intention.

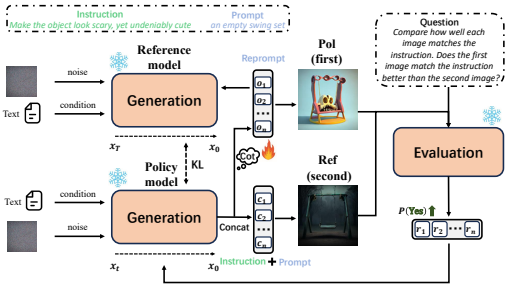


Figure 4: Stage 2: RLMT. The model optimizes its reasoning policy π_θ via a "Think-Generate-Evaluate" loop. The endogenous reward is computed by comparing the reasoned generation (x_{pol}) against a naive baseline (x_{ref}), effectively teaching the model to "think" (reprompt) for superior visual results.

and an initial minimal prompt by $p_0 \in \mathcal{P}$ (e.g., "an empty swing set"). Our objective is to leverage the model's latent capabilities to generate a Reprompt $p \in \mathcal{P}$. Derived from an explicit reasoning path, this reprompt bridges the cognitive gap by translating the user intention of a into concrete descriptors that are executable by the generator. The generation process yields an image $x_{\text{pol}} = G(p)$ that satisfies the instruction a while maintaining the semantic integrity of p_0.

We conceptualize the UMM as comprising multimodal understanding/reasoning parameters θ and visual generation parameters ϕ. The generation process also uses θ to provide semantic conditions for generation. To focus on the "understanding-to-generation" pathway rather than low-level rendering, we freeze the visual generation parameters ϕ and only optimize θ. Here, the generator refers to the conventional visual generation module, rather than the full generative process that also includes reasoning. The model operates in three endogenous functional modes:

- **Generator** $G(\cdot) : \mathcal{P} \to \mathcal{X}$, which maps textual prompts to the image space, with the visual generation parameters ϕ fixed.
- **Evaluator** $E(\cdot|\theta) : \mathcal{X}^2 \times \mathcal{T} \to \{Yes, No\}$, which acts as a pairwise judge that determines whether x_{pol} is preferred over x_{ref} given the context. Here, $\mathcal{T}$ is the textual condition space. $x_{\text{ref}} = G([p_0; a])$ is the baseline.
- **Reprompting Policy** $\pi_\theta : \mathcal{P} \times \mathcal{A} \to \Delta(\mathcal{P})$, which generates the reprompt via reasoning.

Formally, we define the optimization reward R based on the evaluator's pairwise judgment against a naive baseline. For an image pair $\mathbf{x} = (x_{\text{pol}}, x_{\text{ref}})$, we decompose the holistic preference into three complementary aspects (instruction compliance, prompt consistency, and visual quality):

$$R(p; a, p_0) \triangleq \mathbb{I}\left(E(\mathbf{x}; a, p_0) = \text{YES}\right), \tag{1}$$

Specifically, the evaluator assesses whether x_{pol} better follows the instruction a, preserves the semantics of p_0, and maintains overall visual quality. Rather than evaluating these aspects separately, we combine them in a single evaluation prompt and produce one holistic pairwise judgment.

To internalize this reasoning capability, we optimize the policy parameters θ to maximize the expected reward while preventing deviation from the original language distribution:

$$\theta^* = \arg\max_\theta \mathbb{E}_{p\sim\pi_\theta(\cdot|a,p_0)}\Big[R(p; a, p_0)\Big] - \lambda\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \tag{2}$$

where π_{ref} represents the reference policy (the initial model), and D_{KL} acts as a regularization term to ensure the fluency of the generated reprompts.

3.2 Preliminary

Reinforcement Learning with Verifiable Rewards (RLVR) (Wen et al., 2025) optimizes reasoning policies using verifiable rewards, while Reinforcement Learning with Model-rewarded Thinking

(RLMT) (Bhaskar et al., 2025) replaces the verifier with a reward model. SEER combines these two paradigms: RLVR first activates an internal evaluator through verifiable pairwise supervision, and RLMT then uses this evaluator as an endogenous reward model to optimize generative reasoning.

3.3 Visual Instruction Elaboration

To bootstrap generative reasoning efficiently, we construct a compact dataset $\mathcal{D} = \{(a^{(i)}, p_0^{(i)})\}_{i=1}^{300}$. We define this as a proxy task: its purpose is not to cover all possible instructions, but to provide high-leverage samples that bootstrap the model's generative reasoning. We categorize these samples based on the reasoning depth required to bridge the cognitive gap. Simple Instructions focus on direct attribute mapping, covering Material Synthesis, Perspective Shifts, and Semantic Edits (e.g., explicit material synthesis like "carved of ice"). Hard Instructions necessitate deep reasoning to interpret conceptual demands, covering Attribute Reasoning, Constraint Reasoning, and Conceptual Reasoning (e.g., "Make the object look scary, yet undeniably cute"). The details are in Appendix A.1.

The dataset is instantiated into two formats to align with our two-stage training framework. For RLVR, we expand the inputs into quadruplets $\mathcal{D}_{\text{RLVR}} = \{(a_i, p_{0,i}, \mathbf{x}_i, y_i)\}_{i=1}^{300}$. Here, $\mathbf{x}_i = (x_i^+, x_i^-)$ denotes a pair of candidate images collected with minimal human supervision, and $y_i \in \{\text{YES}, \text{NO}\}$ serves as a binary validity label. For RLMT, we utilize the raw inputs $\mathcal{D}_{\text{RLMT}} = \mathcal{D}$ directly, requiring the model to autonomously generate a reasoning chain.

3.4 Self-Evolving Training Framework

We propose SEER, a self-evolving training framework for efficient and automatic endogenous optimization. Instead of relying on an external reward, the framework forms a two-stage loop: RLVR activates the model's latent understanding into a verifiable reward, and RLMT uses this endogenous signal to evolve the reprompting policy.

3.4.1 Stage 1: Reinforcement Learning with Verifiable Rewards (RLVR)

The goal of this stage is to transform the model's multimodal understanding into a reliable endogenous reward signal. We leverage the model to consolidate the three reward dimensions defined in Eq. 1 into a holistic evaluation function $E(x; a, p_0)$ (in Appendix A.3.2), employing Reinforcement Learning with Verifiable Rewards (RLVR) (Figure 3).

Curriculum Learning.

Directly training on visual instructions can lead to unstable gradients. We therefore implement a curriculum to progressively "wake up" the evaluator:

Stage 1: Basic alignment. We train on concrete image–text pairs. The evaluator learns to judge basic visual–semantic alignment, establishing a foundational decision boundary.

Stage 2: Visual Instruction Supervision. We train on quadruplets from $\mathcal{D}_{\text{RLVR}}$. The evaluator must identify which image better satisfies the visual instruction a while preserving p_0, bridging the cognitive gap verifiably.

Pairwise Evaluator Optimization.

We formulate the evaluator's output generation as a sequential decision process and adopt pairwise formulation to mitigate reward hacking (Wang et al., 2025b). Given a query $q = (a, p_0)$ and a candidate image pair $\mathbf{x}$, the evaluation policy π_θ generates a reasoning trace z_i followed by a response (verdict) o_i. We optimize π_θ using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) over the entire trajectory $u_i = [z_i; o_i] = (u_{i,1}, u_{i,2}, \ldots, u_{i,L_i})$. For each query, we sample N trajectories $\{u_i\}_{i=1}^N$ from $\pi_{\theta_{\text{old}}}$. The objective maximizes the advantage-weighted probability per token:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\mathbf{x},\{u_i\}}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{t=1}^{L_i}\left(\min(s_{i,t}^1 A_i, s_{i,t}^2 A_i) - \beta D_{\text{KL}}\left(\pi_\theta(u_{i,t}|u_{i,<t}, q, \mathbf{x}) \| \pi_{\text{ref}}(u_{i,t}|u_{i,<t}, q, \mathbf{x})\right)\right)\right], \tag{3}$$

where $u_{i,t}$ denotes the t-th token of the i-th trajectory. The token-level importance sampling ratios are defined as:

$$s_{i,t}^1 = \frac{\pi_\theta(u_{i,t} \mid u_{i,<t}, q, \mathbf{x})}{\pi_{\theta_{\text{old}}}(u_{i,t} \mid u_{i,<t}, q, \mathbf{x})}, \quad s_{i,t}^2 = \text{clip}\Big(s_{i,t}^1, 1-\epsilon, 1+\epsilon\Big). \tag{4}$$

The advantage A_i is computed at the sequence level by normalizing the scalar rewards within the group:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^N)}{\text{std}(\{r_j\}_{j=1}^N)}. \tag{5}$$

The scalar reward r_i is derived from the binary verification result: $r_i = \mathbb{I}(\text{valid format}) + \mathbb{I}(o_i = y)$.

3.4.2 Stage II: Reinforcement Learning with Model-rewarded Thinking (RLMT)

The goal of this stage is to enhance the model's generative reasoning policy. To achieve this, we employ Reinforcement Learning with Model-rewarded Thinking (RLMT) and leverage the RLVR-tuned evaluator as a reliable endogenous reward model. The evaluator is kept fixed during Stage 2, while the reprompting policy is optimized using its feedback (Figure 4).

Endogenous Relative Reward. To derive a robust scalar reward r_i for each sampled reprompt o_i, we establish a comparative baseline using a naive prompt c_i. This baseline is constructed by directly concatenating the visual instruction a and the initial prompt p_0 (denoted as $c_i = [p_0; a]$), representing a stable and competitive bound. We argue that a successful reprompt must yield visual results superior to this naive concatenation. Accordingly, for a given input (a, p_0), we generate two distinct images:

Reasoned Generation: $x_{\text{pol}} = G(o_i)$, produced using the generated reprompt o_i.

Naive Baseline: $x_{\text{ref}} = G(c_i)$, produced using the direct concatenation c_i.

The total reward r_i aggregates visual preference and format validity. We define the accuracy reward R_{acc} by leveraging the RLVR-tuned evaluator E to compare the images. Following Eq. 1, since the evaluator outputs a binary verdict, the accuracy reward is defined as:

$$R_{\text{acc}} = \mathbb{I}\left(E(x_{\text{pol}}, x_{\text{ref}}; a, p_0) = \textsc{Yes}\right). \tag{6}$$

This assigns a reward of 1 if the reasoned generation x_{pol} is judged superior to the baseline x_{ref}, and 0 otherwise. The final reward is $r_i = R_{\text{acc}} + \mathbb{I}(\text{valid format})$. This mechanism drives the policy to discover reprompts that offer tangible visual improvements over trivial prompt engineering.

Reprompting Policy Optimization. We optimize the reasoning policy π_θ using the identical GRPO objective (Eq. 3). The primary adaptation lies in the input dependency: here, the model generates the reprompt sequence o_i conditioned solely on the textual query $q = (a, p_0)$, without visual context $\mathbf{x}$. Consequently, the probability terms in both the importance sampling ratio and the KL divergence simplify to depend only on the textual context. The token-level ratio becomes:

$$s_{i,t}^1 = \frac{\pi_\theta(u_{i,t} \mid u_{i,<t}, q)}{\pi_{\theta_{\text{old}}}(u_{i,t} \mid u_{i,<t}, q)}. \tag{7}$$

The regularization becomes $D_{\text{KL}}(\pi_\theta(\cdot|q)\|\pi_{\text{ref}}(\cdot|q))$. Substituting the total reward r_i into the advantage calculation aligns the gradient updates with Eq. 2:

$$\theta^* = \arg\max_\theta \mathbb{E}_{p\sim\pi_\theta(\cdot|a,p_0)}\Big[r_i\Big] - \lambda\, D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}}). \tag{8}$$

Crucially, since r_i relies on the contrast between the reasoned output and the naive baseline, this objective biases the policy towards non-trivial reasoning. It forces the model to discover reprompts that yield visual outcomes superior to those of the direct concatenation.

This strategy marks a fundamental shift from traditional RLHF applied to visual generation. Whereas standard RLHF targets the visual generation execution phase (optimizing pixels), SEER targets the generative reasoning process (optimizing prompts). By optimizing the prompts rather than the pixels, SEER effectively teaches the model to "think" before it generates, thereby internalizing the reasoning process and bridging the cognitive gap.

3.4.3 MODEL-SPECIFIC ALIGNMENT

SEER promotes model-specific alignment by optimizing reprompts with feedback from the unified model itself. Unlike external reprompting methods that rely on separate models, SEER uses the shared representation between understanding and generation, allowing the evaluator to provide feedback that is more consistent with the generator's capability. Therefore, the reprompting policy learns to generate descriptions that better satisfy user instructions while remaining executable by the underlying generator. A detailed analysis is provided in Appendix A.10.

4 EXPERIMENTS

In this section, we evaluate SEER's ability to bridge the cognitive gap. We first detail the experimental setup, then present the core results on Visual Instruction Elaboration (evaluating both the internal judge and the generative reasoning), and finally demonstrate that our method preserves the model's general multimodal capabilities.

4.1 EXPERIMENTAL SETUP

4.1.1 BASE MODEL ARCHITECTURE.

We implement our framework on Harmon, a Unified Multimodal Model (UMM) with 1.5B parameters. Harmon adopts a masked autoregressive (MAR) generation paradigm, utilizing a pre-trained MAE (Li et al., 2024) as the visual understanding encoder and a VAE (Kingma & Welling, 2013) as the visual generation tokenizer. During training, we optimize the corresponding multimodal understanding and reasoning parameters in each stage, while keeping the visual generation components frozen. RecA (Xie et al., 2025a) demonstrates that Harmon can achieve reliable reconstruction by conditioning solely on its own understanding embeddings. This latent connectivity is crucial: it implies that the "Internal Evaluator" and "Generator" share a grounded feature space, fulfilling the prerequisite for model-specific alignment derived in Sec. 3.4.3.

4.1.2 IMPLEMENTATION DETAILS.

We conduct all training experiments on 3 NVIDIA A100 (40GB) GPUs using the AdamW (Loshchilov & Hutter, 2017) optimizer. For Stage 1 (RLVR), we employ a staged learning rate schedule: 1×10^{-6} for the Basic Alignment phase and 3×10^{-6} for the Visual Instruction phase. For Stage 2 (RLMT), the learning rate is set to 1×10^{-6}. In the GRPO objective, we set the number of iterations per update to 1, the KL divergence coefficient β to 0.04, and the group size $N = 6$. Both stages are trained for 10 epochs. The entire self-evolution process is highly data-efficient, utilizing only 300 samples.

4.1.3 EVALUATION BENCHMARKS.

To rigorously evaluate our framework, we construct a dedicated Visual Instruction Test set using the same categories defined in Sec. 3.3. We further split it into In-Distribution (ID) (same instructions, different prompts) and Out-of-Distribution (OOD) (different instructions and prompts) subsets to assess generalization in Appendix A.1.5. Additionally, we evaluate general visual understanding capabilities on MME (Fu et al., 2025), POPE (Li et al., 2023b), GQA (Hudson & Manning, 2019), MMMU (Yue et al., 2024), and SEEDBench (Li et al., 2023a), and assess general generation capabilities on GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024).

4.2 MAIN RESULTS: VISUAL INSTRUCTION ELABORATION

4.2.1 EVALUATOR PERFORMANCE (STAGE 1)

We evaluate the internal evaluator on a human-labeled test set of quadruplets. Baselines: (1) Zero-shot: Original Harmon model; (2) Phase 1 Only: Trained on basic alignment only; (3) SEER-Eval: Full two-phase curriculum.

Results. Table 1 presents the accuracy across In-Distribution and Out-Of-Distribution sets, further split into Simple (Simp) and Hard subsets. The Zero-shot model performs near random guessing

Table 1: Evaluator Accuracy on the Visual Instruction Test Set. We report results on both In-Distribution and Out-Of-Distribution splits, further categorized into Simple (Simp) and Hard subsets based on reasoning difficulty.

Model Config	Total	In-Distribution			Out-Of-Distribution		
		Overall	Simp	Hard	Overall	Simp	Hard
Zero-shot	0.41	0.44	0.40	0.47	0.38	0.35	0.40
Phase 1 Only	0.49	0.48	0.45	0.50	0.50	0.65	0.40
SEER-Eval (Ours)	**0.92**	**0.96**	**1.00**	**0.93**	**0.88**	**0.85**	**0.90**

Table 2: Win Ratio of SEER vs. External Reprompting Methods. We report SEER's win rate against each baseline across different reasoning depths. SEER dominates on Hard/Complex tasks while producing efficient reprompts (# Avg. Words).

External Baseline	Total	In-Distribution			Out-Of-Distribution			# Avg. Words
		Overall	Simp	Hard	Overall	Simp	Hard	
vs. SEER-Gen + BeautifulPrompt	0.90	0.89	0.89	0.89	0.90	0.93	0.88	55.13
vs. SEER-Gen + PromptEnhancer	0.75	0.77	0.68	0.83	0.73	0.70	0.74	153.04
vs. SEER-Gen + GPT-5.2	0.68	0.73	0.69	0.76	0.62	0.61	0.63	46.23
vs. SEER-Gen + Gemini3	0.61	0.63	0.51	0.71	0.58	0.56	0.61	33.06
vs. SEER-Gen + Qwen3max	0.65	0.63	0.57	0.68	0.68	0.66	0.70	30.76
SEER (Ours)	-	-	-	-	-	-	-	**22.94**

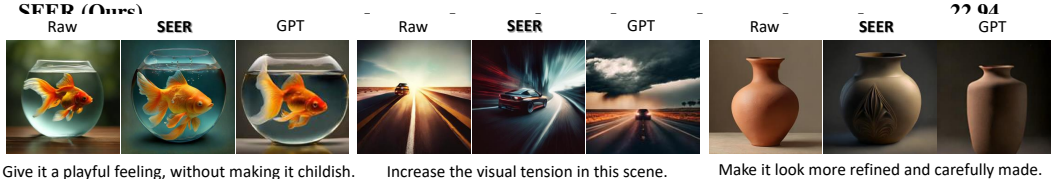


Figure 5: Qualitative Comparison. While the Base Model (Left) fails to execute visual instructions due to the cognitive gap, and External Reprompters (Right) cause representation mismatch, SEER (Center) generates reprompts that match the generator's priors, yielding superior visual fidelity.

(0.41), confirming that latent understanding does not automatically translate to verification capability. Phase 1 Only improves basic alignment but stalls on instruction tasks (0.49). In contrast, SEER-Eval achieves a remarkable 0.92 overall accuracy. Notably, it maintains 0.90 on OOD Hard instructions. This confirms that the "Compact Proxy Task" effectively bootstraps the latent evaluation capability, enabling the model to generalize to unseen concepts without massive-scale supervision. Furthermore, we perform a data scale ablation on the evaluator in Appendix A.4, demonstrating that accuracy jumps non-linearly from 0.60 to 0.92 as sample size scales from 50 to 300.

4.2.2 Generative Reasoning Performance (Stage 2)

We evaluate the final generation quality via blind pairwise human evaluation.

Metric: Human Win Ratio. To quantify performance, we conduct blind pairwise comparisons. Annotators are presented with an image from SEER and a baseline, selecting the one that better satisfies the visual instruction a while preserving the semantic subject of p_0. A Win Ratio > 0.5 indicates that SEER is preferred (details in the Appendix A.2). We measure substantial inter-annotator agreement (Fleiss' $\kappa = 0.72$), with further statistical analysis provided in Appendix A.2.

Comparison with External Reprompters. We benchmark SEER against two categories: (1) Specialized Reprompters: BeautifulPrompt (Cao et al., 2023), PromptEnhancer (Wang et al., 2025a); and (2) SOTA MLLMs: GPT-5.2 (OpenAI, 2025), Gemini3 (Google DeepMind, 2025), Qwen3max (Alibaba, 2025). For a fair comparison, we substitute SEER's reprompting policy with these external models, feeding their generated reprompts directly into SEER's generator. Thus, all reprompting methods are evaluated with the same generator, isolating the effect of the reprompt source.

Results. As shown in Table 2, SEER consistently outperforms both groups. It surpasses specialized reprompters (0.90 against BeautifulPrompt), supporting our hypothesis that representation mismatch can reduce the effectiveness of external reprompts. SOTA MLLMs perform competitively (0.61

Table 3: Human Win Ratio of SEER (Ours) against various baselines. A score > 0.5 indicates SEER is preferred. Results are reported across In-Distribution and Out-Of-Distribution splits, further categorized into Simple (Simp) and Hard subsets.

Opponent Model	Total	In-Distribution			Out-Of-Distribution		
		Overall	Simp	Hard	Overall	Simp	Hard
vs. Base (CoT)	0.85	0.85	0.78	0.90	0.85	0.87	0.84
vs. Bagel	0.75	0.77	0.64	0.86	0.73	0.63	0.79
vs. Bagel-Think	0.69	0.77	0.63	0.86	0.62	0.55	0.66
vs. Blip3-o	0.81	0.80	0.76	0.83	0.83	0.93	0.76
vs. Show-o2	0.81	0.85	0.78	0.89	0.77	0.73	0.80

Table 4: General Generation Capabilities on GenEval and DPG-Bench. Note: For these simple prompts, we disable the reprompting mechanism to test the direct generation quality. SEER maintains or improves fundamental compositional capabilities.

Model	GenEval							DPG-Bench (Score)
	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall	
Harmon (Base)	0.99	0.86	0.68	0.85	0.43	0.48	0.72	80.5
Stage 1 Only	0.99	0.87	0.68	0.86	0.44	0.48	0.72	80.9
Stage 2 (SEER)	**1.00**	**0.88**	**0.70**	**0.87**	**0.44**	**0.52**	**0.74**	**81.1**

against Gemini3), benefiting from stronger understanding knowledge; however, SEER still achieves superior performance with significantly fewer words (22.94 on average). This efficient reprompting behavior, together with the higher win ratio, further supports our model-specific alignment hypothesis in Sec. 3.4.3. Notably, SEER's advantage is often amplified on Hard Instructions (e.g., vs PromptEnhancer: 0.74 on Hard vs 0.70 on Simple). Figure 5 provides qualitative examples.

Comparison with UMMs. We further compare SEER (1.5B) with state-of-the-art UMMs: Harmon (Base), Bagel (14B) (Deng et al., 2025), Bagel-Think (CoT), Blip3-o (8B) (Chen et al., 2025a), and Show-o2 (7B) (Xie et al., 2025b).

Results. Table 3 reports the Human Win Ratio of SEER against each baseline. First, SEER achieves a dominant win ratio (e.g., 0.85) against the Harmon Base, confirming that Endogenous Reprompting effectively bridges the cognitive gap. Second, despite its smaller parameter size (1.5B), SEER remains competitive against significantly larger models. This competitiveness is largely driven by SEER's robustness on Hard Instructions (e.g., vs Bagel: 0.79 on Hard vs 0.63 on Simple). Critically, this high alignment with human preferences indicates that our Internal Evaluator (Stage 1) provides effective guidance for the Reprompting Policy (Stage 2) to generate self-aligned reprompts. Automatic metrics and validation on Show-o2 further support SEER's effectiveness (Appendix A.6-A.7).

4.3 General Capabilities

We evaluate whether SEER preserves general multimodal capabilities on standard understanding and generation benchmarks. On MME, POPE, GQA, MMMU, and SEEDBench, SEER remains stable compared with Harmon (e.g., MME: 1155 $\rightarrow$ 1179), with detailed results provided in Appendix A.5. For generation, we disable reprompting and directly evaluate on GenEval and DPG-Bench. As shown in Table 4, SEER slightly improves performance (GenEval: 0.72 $\rightarrow$ 0.74; DPG-Bench: 80.5 $\rightarrow$ 81.1), showing that our reasoning optimization does not sacrifice general multimodal capabilities.

5 Conclusion

We address the cognitive gap in UMMs by introducing SEER, a self-evolving framework that transforms passive understanding into active reasoning. Through a two-stage endogenous loop (RLVR and RLMT), SEER bootstraps the model's reprompting capability using only 300 samples via Visual Instruction Elaboration. Experiments confirm that SEER outperforms baselines in evaluation accuracy, reprompting efficiency, and generation quality while preserving general capabilities. Crucially, our approach promotes model-specific alignment, enabling reasoning to produce effective triggers for the generator's generative priors. This work highlights a new direction: shifting focus from optimizing execution to evolving cognitive reasoning.

AI USE STATEMENT

In this work, we used GPT-5 for semantic consistency filtering during dataset construction, as described in Appendix A.1.2. We did not use generative AI tools for other research tasks requiring disclosure, and the remaining required disclosure tasks are not applicable to this work. We did not use generative AI tools for tasks with recommended disclosure. All samples retained after AI-assisted filtering were further subject to human verification. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not use private data. Human annotators were used for pairwise evaluation and sparse verification of synthetic samples, while the remaining evaluations use publicly available benchmarks and our synthetic visual instruction dataset. Potential misuse follows the general risks associated with generative image models. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

Implementation details and hyperparameters are provided in Sec. 4.1.2, while the data construction pipeline, evaluation protocol, and prompt templates are described in Appendices A.1–A.3. An anonymous code repository is provided in Appendix A.1. We will release the dataset and training/evaluation scripts upon publication.

REFERENCES

Alibaba. Qwen3 max. `https://qwen.ai/`, 2025. Trillion-parameter AI model (Qwen3 Max) from Alibaba Cloud, accessed Jan. 2026.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Adithya Bhaskar, Xi Ye, and Danqi Chen. Language models that think, chat better. *arXiv preprint arXiv:2509.20357*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. Beautifulprompt: Towards automatic prompt engineering for text-to-image synthesis. *arXiv preprint arXiv:2311.06752*, 2023.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

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

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

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

Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023.

Zhen Fang, Ruiyan Han, XinYu Sun, Yuchen Ma, Ziheng Wang, Yu Zeng, Zehui Chen, Lin Chen, Wenxuan Huang, Wei-Jie Xu, et al. Unicorn: Towards self-improving unified multimodal models through self-generated supervision. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13638–13669, 2026.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Yuan Gong, Xionghui Wang, Jie Wu, Shiyin Wang, Yitong Wang, and Xinglong Wu. Onereward: Unified mask-guided image generation via multi-task human preference learning. *arXiv preprint arXiv:2508.21066*, 2025.

Google DeepMind. Gemini 3. `https://gemini.google.com/`, 2025. Next-generation large multimodal language model from Google, accessed Jan. 2026.

Yujin Han, Hao Chen, Andi Han, Zhiheng Wang, Xinyu Liu, Shiwei Zhang, Difan Zou, et al. Turning internal gap into self-improvement: Promoting the generation-understanding unification in mllms. In *International Conference on Learning Representations*, volume 2026, pp. 136501–136534, 2026.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Weiyang Jin, Yuwei Niu, Jiaqi Liao, Chengqi Duan, Aoxue Li, Shenghua Gao, and Xihui Liu. Srum: Fine-grained self-rewarding for unified multimodal models. *arXiv preprint arXiv:2510.12784*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pp. 159–174, 1977.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37: 56424–56445, 2024.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023b.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

OpenAI. Gpt-5.2. https://openai.com, 2025. Large multimodal language model, accessed Jan. 2026.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

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

Linqing Wang, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Jiale Tao, Qixun Wang, Ruihuang Li, Comi Chen, et al. Promptenhancer: A simple approach to enhance text-to-image models via chain-of-thought prompt rewriting. *arXiv preprint arXiv:2509.04545*, 2025a.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024a.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025b.

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

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12966–12977, 2025a.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, et al. Rewarddance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025b.

Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025c.

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

Ji Xie, Trevor Darrell, Luke Zettlemoyer, and XuDong Wang. Reconstruction alignment improves unified multimodal models. *arXiv preprint arXiv:2509.07295*, 2025a.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025b.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 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*, pp. 9556–9567, 2024.

Chuyang Zhao, Yuxing Song, Wenhao Wang, Haocheng Feng, Errui Ding, Yifan Sun, Xinyan Xiao, and Jingdong Wang. Monoformer: One transformer for both diffusion and autoregression. *arXiv preprint arXiv:2409.16280*, 2024.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A APPENDIX

A.1 DATASET CONSTRUCTION DETAILS: VISUAL INSTRUCTION ELABORATION

To effectively bootstrap the model's generative reasoning, we constructed a compact but high-leverage dataset for the Visual Instruction Elaboration task. The dataset is structured along two orthogonal dimensions of reasoning depth, comprising six distinct semantic categories. **Anonymous Code Link:** https://anonymous.4open.science/r/seer-anonymous-22D4/

A.1.1 DATA CATEGORIZATION AND EXAMPLES

We categorize instructions into Simple and Hard groups. Simple Instructions focus on direct attribute mapping, while Hard Instructions necessitate deep inferential reasoning. Table 5 provides specific definitions and examples for each category.

Table 5: Detailed Categories of the Visual Instruction Elaboration Dataset. We cover 6 categories ranging from explicit synthesis to abstract reasoning.

Difficulty	Category	Description & Example
Simple	**Material Synthesis**	Explicitly changing the texture or material of the object. *Ex: "Transform it so it is carved entirely out of ice."*
	Perspective Shifts	Altering the camera angle or visual style without changing semantics. *Ex: "Show it through a grainy surveillance footage."*
	Semantic Edits	Adding explicit accessories or modifying basic attributes. *Ex: "Place a golden crown on the object's head or top."*
Hard	**Attribute Reasoning**	Inferring implicit visual details from underspecified descriptions. *Ex: "Make it look more refined and carefully made."*
	Constraint Reasoning	Resolving semantic conflicts or balancing paradoxical constraints. *Ex: "Make the object look scary, yet undeniably cute."*
	Conceptual Reasoning	Grounding high-level visual concepts into concrete visual elements. *Ex: "Make this image more memorable."*

A.1.2 DATA COLLECTION PIPELINE (TRAINING)

We constructed the training set using a scalable, minimal-supervision strategy to balance the quality and efficiency of self-evolution. The pipeline consists of three steps:

1. **Candidate Generation:** We generate a diverse pool of 600 candidate samples (spanning 6 categories with 10 instructions per category).
2. **Image-Based Feasibility Filtering:** To ensure the training instructions were visually realizable given the UMM's base capability, we generated 600 candidate images from the initial pairs. We then applied a dual-filter mechanism to evaluate the alignment between the generated image, the instruction, and the prompt: (1) Automated Check: First, GPT-5 performed a semantic consistency check to reject obviously low-quality or misaligned generations. (2) Sparse Human Verification: Second, for the remaining candidates, we conducted sparse human verification, where annotators quickly validated the fine-grained alignment. Only the 300 pairs that passed both filters were retained.
3. **Final Selection:** From the filtered pool, we retained the top 300 high-quality (a, p_0) pairs for training. This process ensures the dataset is accurate enough for RLVR while minimizing human annotation costs.

A.1.3 TEST SET CONFIGURATION

To rigorously assess the capabilities of SEER, we constructed a dedicated test set completely separate from the training data. Crucially, unlike the training set which was filtered for visual realizability to facilitate learning, the test set was curated solely based on semantic coverage without considering the

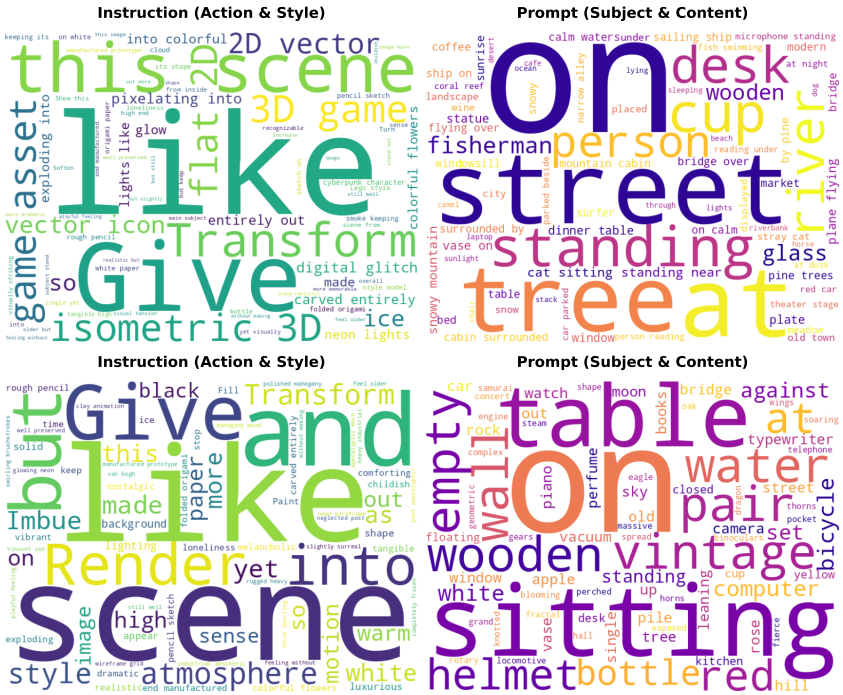


Figure 6: Lexical Distribution Shift Analysis. We visualize the word clouds for Instructions (Left) and Prompts (Right) across the Training (Top) and Test (Below) sets. Prompts: The significant vocabulary shift (e.g., "street" → "table") ensures rigorous testing of object generalization. Instructions: While structural triggers overlap to maintain task consistency, the Test set introduces novel vocabulary (OOD concepts), challenging the model's zero-shot reasoning capabilities.

model's base capability. This ensures a fair, unbiased, and comprehensive evaluation of the model's true boundaries.

- **Scale:** We explicitly selected 40 instructions.
- **Distribution Split:**
  - In-Distribution (ID): Instructions seen during training but combined with entirely new prompts p_0 to test combinatorial generalization.
  - Out-Of-Distribution (OOD): Instructions representing entirely unseen semantic concepts (e.g., new materials or abstract vibes) to test zero-shot transfer capabilities.
  - The ratio of ID to OOD is 1:1 (20 vs. 20).
- **Difficulty Split:** The ratio of Simple to Hard instructions is 2:3, emphasizing the evaluation of complex reasoning capabilities on harder tasks.

A.1.4 Training Curriculum and Negative Sampling

For the RLVR (Evaluator Training) stage, we utilize the 300 training pairs (a, p_0) constructed in Sec. A.1.2. We implement a two-phase curriculum to progressively activate the evaluators:

Curriculum 1: Basic Alignment (Warm-up). To teach the evaluator basic image-text alignment, we use the initial prompt p_0 from the training set.

- **Positive Sample (x_{c1}^+):** The image generated directly using p_0.
- **Negative Sample (x_{c1}^-):** An image generated by randomly masking or adding the object or scene in p_0, creating a semantic mismatch.

Curriculum 2: Instruction Supervision (Main). This phase teaches the model to verify the visual instruction a.

- **Positive Sample (x_{c2}^{+}):** The verified ground-truth image from the training set (sampled and filtered in the data collection pipeline).
- **Negative Sample (x_{c2}^{-}):** We directly reuse the Positive Sample from Curriculum 1 (x_{c1}^{+}). Since this image reflects only p_0 without the instruction a, it serves as a hard negative. For each pair, we randomly swap the image order and update the Yes/No label accordingly to avoid position bias.

The evaluator learns to prefer x_{c2}^{+} over x_{c2}^{-}. Crucially, we maintain a clear distinction in data curation: while these training samples rely on minimal supervision for efficiency, the test set (used for Table 1) underwent rigorous multi-round human review to ensure the Ground Truth aligns with human intuition.

A.1.5 Distributional Analysis: Train vs. Test.

To evaluate the model's generalization capabilities, we analyze the lexical overlap between the training and test splits, as visualized in Figure 6 and Table 6.

- **Prompt Divergence (Object Generalization):** As shown in the Prompt word clouds (Right), while both sets demonstrate comprehensive semantic diversity, their vocabularies are markedly distinct. The training set covers a wide array of entities (e.g., "street", "standing", "river"), whereas the test set spans an equally diverse but disjoint set of subjects (e.g., "table", "wooden", "vintage"). This disjoint yet diverse distribution ensures that the model cannot simply memorize object-attribute pairs; it must learn to apply instruction-guided modifications to entirely new semantic subjects.
- **Instruction Expansion (Concept Generalization):** In the Instruction clouds (Left), while structural keywords (e.g., "Transform", "like") overlap to define the task format, the test set introduces a broader spectrum of semantic demands. It covers not only the seen In-Distribution (ID) types but also expands into Out-Of-Distribution (OOD) concepts (e.g., specific artistic styles or abstract atmospheres not present in training), forcing the model to perform true generative reasoning rather than pattern matching.

Table 6: Distributional Shift Analysis: ID vs. OOD. We provide concrete examples demonstrating the semantic gap between In-Distribution (Seen concepts) and Out-of-Distribution (Unseen concepts) across all 6 reasoning categories.

Difficulty	Category	In-Distribution (ID) Example *(Concept seen during training)*	Out-Of-Distribution (OOD) Example *(Novel concept unseen during training)*
Simple	Material Synthesis	*"Transform it so it is carved entirely out of **ice**."*	*"Transform it so it is made of **polished mahogany wood**."*
	Perspective Shifts	*"Show it through a **grainy surveillance footage**."*	*"Render it in the style of **8-bit retro pixel art**."*
	Semantic Edits	*"Place a **golden crown** on the object's head."*	*"Transform the object into a **die-cut sticker with a white border**."*
Hard	Inferential Reasoning	*"Make it look more **refined and carefully made**."*	*"Give it a **rugged, heavy industrial aesthetic**."*
	Constraint Reasoning	*"Make the object look **scary, yet undeniably cute**."*	*"Render it as **cheap plastic**, yet make it appear **precious and luxurious**."*
	Conceptual Reasoning	*"Make this image more **memorable**."*	*"Make this image convey a sense of **impending danger**."*

A.2 Human Evaluation Protocol

To calculate the Human Win Ratio, we adhered to a strict blind pairwise protocol:

- **Annotators:** 6 independent human evaluators.
- **Procedure:** For each test case, the evaluator is presented with two images: one from SEER and one from a Baseline model. The order of images is randomized to prevent position bias.
- **Criterion:** "Which image better satisfies the instruction 'a' while preserving the subject 'p_0'?"
- **Scoring:**
  - If SEER is chosen: +1 point.

  - If Baseline is chosen: 0 points.
  - If Tie: 0.5 points.
- **Final Metric:** The final Win Ratio is the average score across all annotators and all cases.
- **Statistical Reliability:** To ensure evaluation robustness, we measured inter-annotator agreement and correlation with automated metrics:
  - *Inter-annotator Agreement:* Fleiss' $\kappa = 0.72$ across the 6 evaluators, indicating substantial agreement according to the Landis & Koch scale (Landis & Koch, 1977).
  - *Significance Testing:* For all win ratios, we conducted two-sided binomial tests over non-tied comparisons. Ties were excluded from the significance test. SEER's gains over Harmon (Base) are highly significant ($p < 0.001$), and all baseline comparisons remain statistically significant ($p < 0.05$).
  - *Correlation with Automatic Metrics:* We calculated the Pearson correlation coefficient between the human preference scores and automatic evaluation metrics. We observed a strong positive correlation with both CLIP Score ($r = 0.748$, $p = 0.013$) and ImageReward ($r = 0.705$, $p = 0.023$), showing consistent trends between human judgments and these automatic metrics.

A.3 Prompt Templates

We utilized structured system prompts to enforce the Chain-of-Thought (CoT) reasoning format during both training and inference.

A.3.1 System Prompt for Reasoning

The following system prompt guides the model to explicitize its reasoning process before generating the final reprompt.

```
A conversation between User and Assistant.
The user asks a question, and the Assistant solves it.
The assistant first thinks about the reasoning process in the mind
and then provides the user with the answer.
The reasoning process and answer are enclosed within <think> </think>
and <answer> </answer> tags, respectively, i.e.,
<think> reasoning process here </think>
<answer> answer here </answer>
```

A.3.2 Prompt for Pairwise Evaluation (Stage 1)

During RLVR training and inference, we leverage the VLM's inherent reasoning capabilities to consolidate the three reward dimensions (defined in Eq. 1) into a single holistic judgment. As shown in the template below, the instruction explicitly fuses Compliance, Consistency, and Visual Quality into a unified comparison criterion.

```
Here is the first image. <image>
Here is the second image. <image>
User Instruction: {instruction}
Base Prompt: {prompt}
Compare the images. The goal is to apply the Instruction to the Base
Subject.
You MUST prefer the image that correctly follows the instruction
(e.g., material/perspective change) while keeping the subject
recognizable and having better quality.
Does the first image match the instruction better than the second
image?
First output the thinking process in <think> </think> tags and then
output the final answer with only 'Yes' or 'No' in <answer> </answer>
tags.
```

A.3.3 Prompt for Endogenous Reprompting (Stage 2)

During RLMT training and inference, the generator uses the following template to generate reprompts:

```
Rewrite the base prompt to follow the instruction while preserving
the identity of the core subject.
User Instruction: instruction
Base Prompt: prompt
1. Identify Subject: Keep the main object in the base prompt as the
central focus.
2. Translate to Visuals: Convert the instruction into concrete
visual descriptors (e.g., specific style, lighting, texture, or
composition details).
3. Rewrite: Write the final prompt integrating these new details.
First output the reasoning in <think> </think> tags, then output the
final image generation prompt in <answer> </answer> tags.
```

A.4 Data Scale Ablation

To evaluate the impact of training sample size on the internal evaluator's accuracy, we trained the evaluator using subsets of the compact proxy task dataset with different sizes. As shown in Table 7, the evaluator accuracy jumps non-linearly. With only 50 or 150 samples, the evaluator struggles to capture complex verification logic. However, scaling to 300 samples boosts the verification accuracy to 0.92, providing the reliable reward signal required to guide the policy.

Table 7: Data Scale Ablation. Verification accuracy across different training sample counts on ID and OOD splits.

Training Samples	ID Accuracy	OOD Accuracy	Overall Accuracy
50	0.61	0.59	0.60
150	0.69	0.65	0.67
300	0.93	0.91	0.92

A.5 General Multimodal Understanding

To verify that SEER preserves general multimodal understanding capabilities, we evaluate the model on MME, POPE, GQA, MMMU, and SEEDBench. As shown in Table 8, SEER remains stable compared with the Harmon base model, with only minor fluctuations across benchmarks. In particular, MME improves from 1155 to 1179, while the other benchmarks remain comparable. These results show that the two-stage training preserves the model's general understanding capabilities.

A.6 Objective Metrics Comparison

To complement the human preference evaluations, we measured automatic quality indicators. Table 9 and Table 10 report the CLIP Score and Image-Reward across all baselines. SEER outperforms both specialized external reprompters and SOTA unified multimodal models. It achieves a CLIP Score of 28.63 and an Image-Reward of 1.13, which correlates strongly with the human judgments discussed in Section A.2.

A.7 Cross-Architecture Validation on Show-o2

To verify the generalizability of SEER across architectures, we applied our two-stage training pipeline to the 1.5B variant of Show-o2. Show-o2 uses an autoregressive next-token prediction architecture with discrete diffusion, which is fundamentally different from Harmon's masked autoregressive (MAR) architecture. We used the same 300 proxy task samples and identical training hyperparameters.

As shown in Table 11, SEER significantly improves Show-o2's generation quality. It achieves a 78.0% win ratio over the base Show-o2 model in blind human preference evaluation. Automatic metrics

Table 8: General Multimodal Understanding Benchmarks. We report results across progressive training stages. SEER (Stage 2) maintains comparable understanding capabilities to the Harmon base model.

Model	MME	$POPE_{Acc}$	$POPE_{F1}$	GQA	MMMU	SEED
Harmon (Base)	1155	83.8	83.9	58.8	34.7	65.2
Phase 1 Only	1172	83.7	83.9	58.9	34.8	65.2
Stage 1 (RLVR)	1179	84.0	84.3	58.9	35.2	64.4
Stage 2 (SEER)	1179	83.8	84.1	58.9	35.1	65.4

Table 9: Objective Metrics Comparison against External Reprompters.

Model	**CLIP Score (↑)**	**Image-Reward (↑)**
BeautifulPrompt	23.44	-0.18
PromptEnhancer	27.39	0.96
GPT-5.2	27.71	0.98
Gemini3	28.32	0.99
Qwen3max	28.34	1.10
SEER (Ours)	**28.63**	**1.13**

also show consistent improvements: CLIP Score increases from 27.10 to 28.48, and Image-Reward rises from 0.52 to 0.89. This shows that endogenous alignment can generalize across different model backbones without model-specific architecture tuning.

A.8 Qualitative Visualizations

A.8.1 Consistency of Reprompting (Ablation)

In Figure 7, we visualize the generation outcomes when feeding reprompts from different sources (e.g., GPT-5.2, PromptEnhancer, vs. SEER) into the fixed SEER-Gen. This qualitative ablation demonstrates that while external reprompts may be linguistically rich, they often cause the generator to miss attributes due to representation mismatch. In contrast, SEER's self-aligned reprompts effectively trigger the generator's priors, confirming the benefit of model-specific alignment.

A.8.2 Comparison with SOTA UMMs

Figure 8 presents a gallery of comparisons between SEER and baselines (Bagel, Show-o2) on visual instructions. By leveraging generative reasoning, SEER faithfully executes the user's intent across diverse scenarios, whereas baselines often suffer from semantic collapse, attribute neglect, or literal interpretation failures.

A.8.3 Fine-grained Analysis of the Cognitive Gap

To precisely characterize the cognitive gap, we categorize the typical failure modes of UMMs into three distinct types, as illustrated in Figure 9. These failures represent the specific behaviors our endogenous reward model aims to penalize, driving the policy towards more effective reasoning.

- **Type I: Naive Concatenation (Laziness).** The model simply appends the instruction text or its synonyms to the prompt without semantic integration (e.g., Prompt: "A cat", Output: "A cat, make it cute"). This reflects a lack of active planning, resulting in images that fail to visualize the intended change.
- **Type II: Semantic Misinterpretation (Hallucination).** The model misunderstands the instruction's intent, translating it into unrelated or conflicting visual features (e.g., interpreting "frozen in time" as "covered in ice" instead of "motionless"). This leads to visual hallucinations that diverge from the user's goal.

Table 10: Objective Metrics Comparison against SOTA UMMs.

Model	CLIP Score (↑)	Image-Reward (↑)
Harmon (Base) (1.5B)	25.97	0.73
Bagel (14B)	27.93	1.07
Bagel-Think (CoT) (14B)	27.98	1.10
Blip3-o (8B)	27.50	0.75
Show-o2 (7B)	28.33	1.06
SEER (Ours) (1.5B)	**28.63**	**1.13**

Table 11: Cross-Architecture Validation Results on Show-o2.

Method	CLIP Score (↑)	Image-Reward (↑)	Human Win Ratio vs. Base (↑)
Show-o2 (Base)	27.10	0.52	—
Show-o2 + SEER	28.48	0.89	78.0%

- **Type III: Instruction Neglect (Omission).** The model correctly identifies the subject (p_0) but completely ignores the visual modifier (a). The generated image depicts the base object perfectly but lacks the requested style or attribute. This is the most common manifestation of the gap in "Hard" tasks.

Failure Mode Quantitative Distribution To quantify the mitigation of the cognitive gap, we measured the distribution of the three failure modes on 100 complex test instructions. As reported in Table 12, SEER reduces Type I (Laziness) from 25% to 3%, Type II (Hallucination) from 6% to 1%, and Type III (Omission) from 18% to 2%. This massive reduction in attribute omission and naive copying demonstrates how explicit generative reasoning bridges the understanding-to-generation gap.

Table 12: Quantitative Failure Mode Distribution (N=100).

Model	Success Rate (↑)	Type I: Laziness	Type II: Hallucination	Type III: Omission
Harmon (Base)	51%	25%	6%	18%
SEER (Ours)	**94%**	**3%**	**1%**	**2%**

Optimization Mechanism. During the RLMT stage, the policy explores various reasoning paths. Some generated CoT traces may still lead to these failure modes. Since the internal evaluator assigns low rewards to the images resulting from these ineffective CoTs (often scoring them lower than the naive baseline), the optimization process naturally suppresses the probability of such reasoning traces. Simultaneously, it reinforces the effective CoT paths that successfully bridge the gap, thereby distilling the model's reasoning capability to produce high-fidelity generation.

A.9 Second-Round Self-Evolution and Hard Dataset Evaluation

To investigate the limits of self-evolution, we perform a second-round training. We use first-round generations as positive samples (x^{+}_{round2}) and treat the original positives as hard negatives (x^{-}_{round2}). We then construct a more challenging test set, SEER-Eval2, to evaluate the fine-grained discrimination of the evaluators.

Table 13 reports the human-model alignment accuracy on the hard SEER-Eval2 test set. Due to high similarity between candidates, SEER-Eval (Round 1) drops to 59.0% accuracy. In contrast, SEER-Eval2 (Round 2) successfully distinguishes the subtle differences, achieving a robust 76.0% overall accuracy (80.0% on hard instructions).

Additionally, Table 14 presents the pairwise win ratio of the second-round generation policy over the first-round policy. The second-round policy consistently outperforms the first-round policy, achieving a 62.25% overall win rate. This confirms that finer-grained evaluator feedback successfully guides the generation policy to produce superior visual-semantic alignment. However, because the underlying

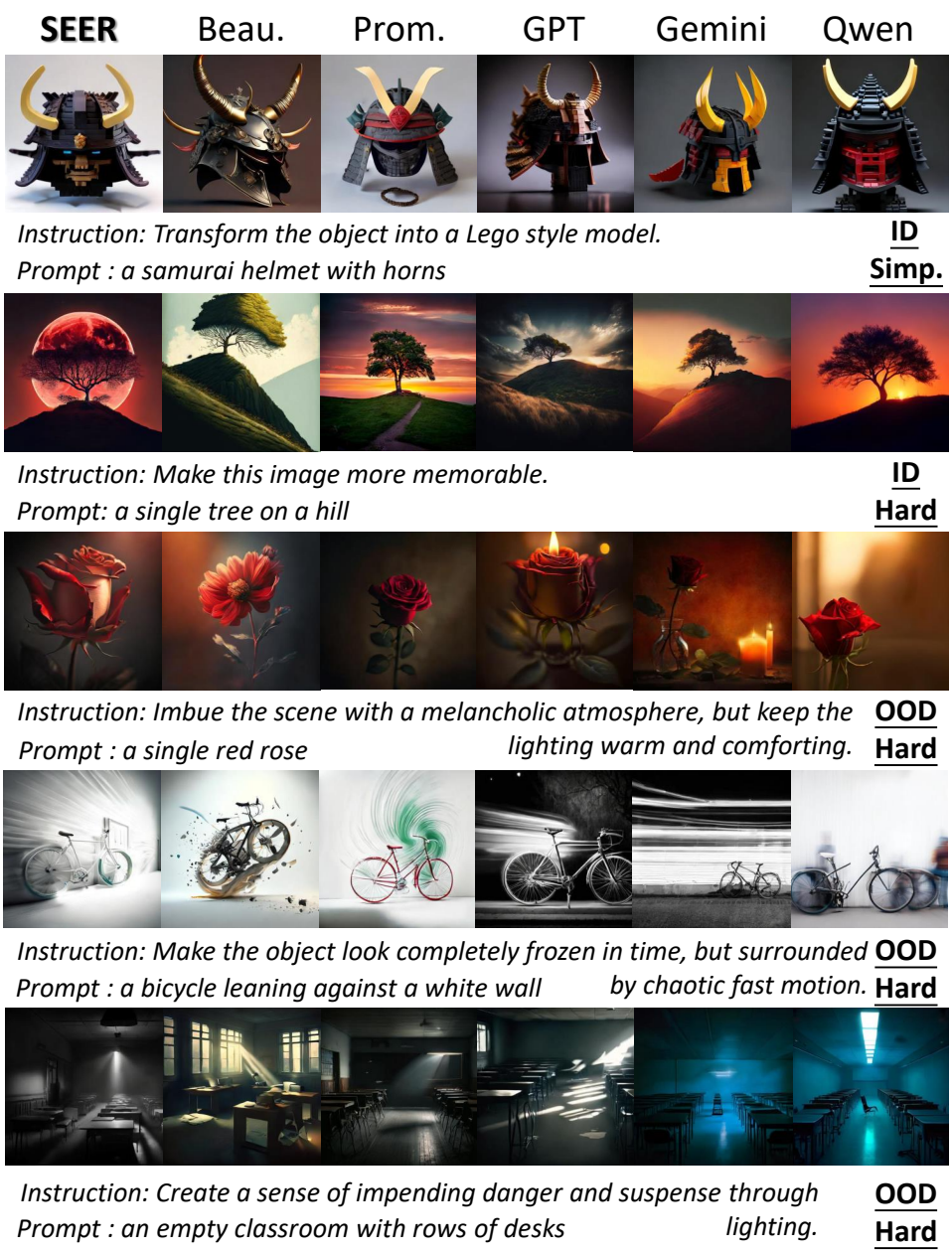


Figure 7: Qualitative Ablation: Impact of Reprompter Source. We verify model-specific alignment by feeding different reprompts into the fixed SEER-Gen. While external reprompters (Middle/Right) often suffer from Instruction Neglect, rendering the prompt object literally but failing to trigger the requested instruction shift (e.g., Lego, Fast Motion). SEER (Left) successfully concretizes visual instructions into effective visual descriptions.

image generator remains unoptimized throughout the loop, the benefits of purely reasoning-level optimization begin to show diminishing marginal returns, as performance gradually approaches the physical capacity ceiling of the frozen generator.

A.10 Analysis of Model-Specific Alignment

A key property of SEER is that it promotes model-specific alignment. Let $\mathcal{Z}_{\text{gen}}$ denote the latent content space of generative priors that the generator G can reliably realize. By combining disjoint models, prior methods suffer from representation mismatch: the prompter generates descriptions

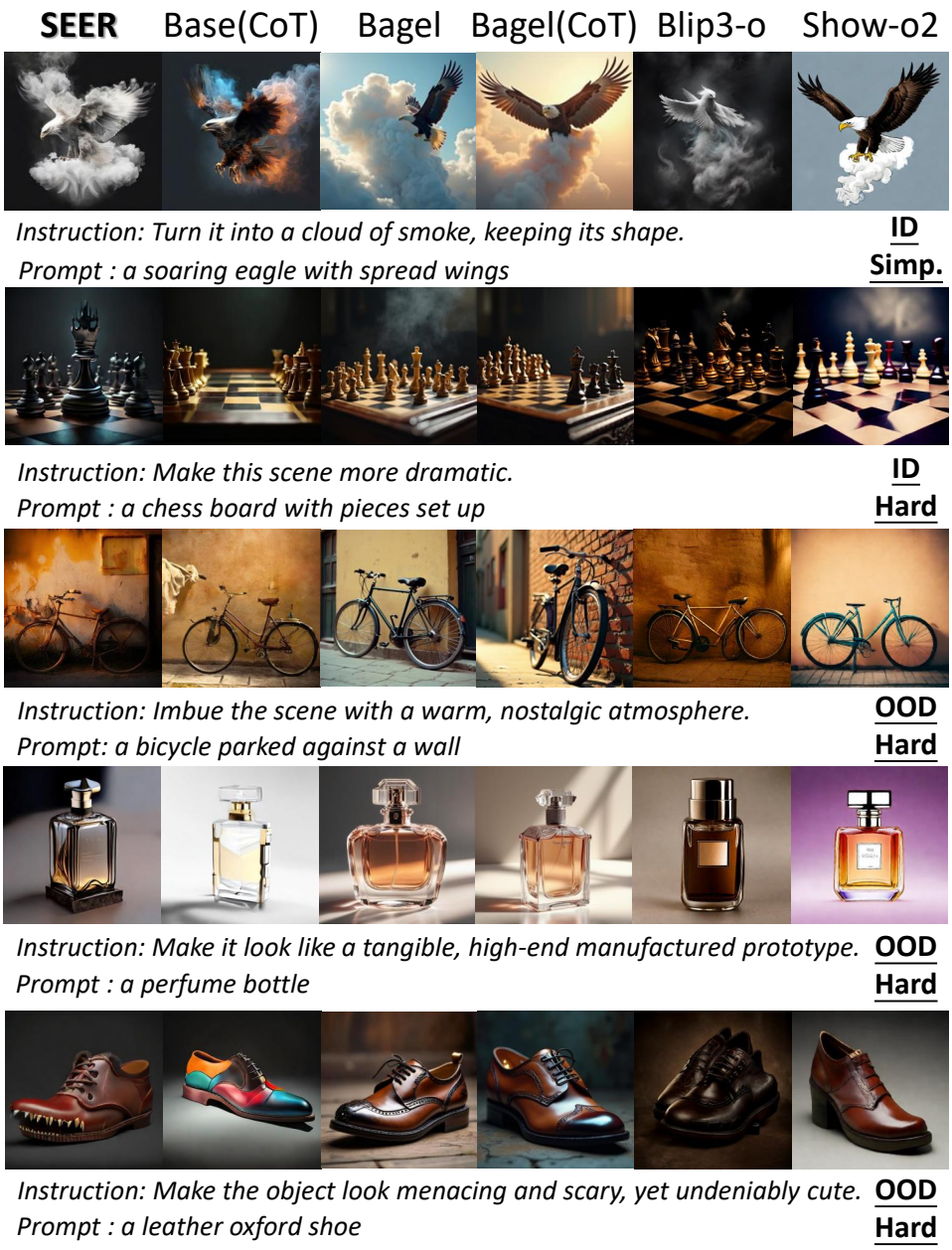


Figure 8: Visual Benchmarking against SOTA UMMs. Baselines (Middle/Right) frequently suffer from literal interpretation, ignoring visual instructions (e.g., drawing a normal eagle instead of "smoke", or a normal shoe instead of "scary and cute"). SEER (Left) demonstrates superior generative reasoning, successfully fusing the user intent with the subject.

without knowing if the generator can actually visualize them, and even when tuned with an external reward R_{ext}, they optimize a signal that fails to accurately assess the generator's capabilities:

$$p^*_{\text{ext}} = \arg\max_p R_{\text{ext}}(G(p), a). \tag{9}$$

Consequently, the generated reprompts p^*_{ext} often drift away from $\mathcal{Z}_{\text{gen}}$, leading to descriptions that are linguistically valid but visually unrealizable. In contrast, Endogenous Reprompting resolves this mismatch by leveraging the intrinsic representational consistency of UMMs. While Eq. 2 explicitly optimizes a relative preference, we provide an intuitive interpretation under the Bradley-Terry assumption (Bradley & Terry, 1952). For this interpretation, we model the underlying pairwise

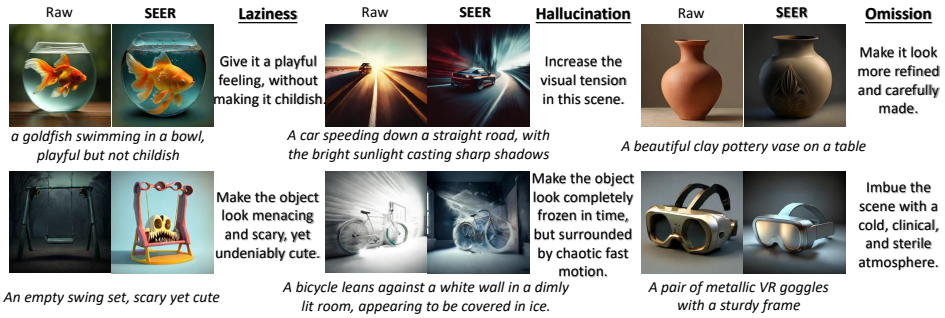


Figure 9: Visualization of Cognitive Gap Failure Modes. We compare the Raw baseline with SEER across three failure types: (I) Laziness (naive concatenation), (II) Hallucination (semantic misinterpretation), and (III) Omission (instruction neglect). The text below each image pair shows the reprompt generated by the Raw baseline. SEER better executes complex instructions through endogenous reasoning.

Table 13: Evaluator Accuracy on the Hard Test Set (SEER-Eval2). We report human-model alignment accuracy of SEER-Eval (Round 1) and SEER-Eval2 (Round 2) across progressive difficulty intervals.

Evaluator Version	**Overall 0–99**	**Simple 0–49**	**Hard 50–99**	**Sub-Intervals 0–19**	**20–49**	**50–69**	**70–99**
SEER-Eval (Round 1)	59.0%	62.0%	56.0%	70.0%	56.7%	55.0%	56.7%
SEER-Eval2 (Round 2)	**76.0%**	**72.0%**	**80.0%**	**85.0%**	**63.3%**	**80.0%**	**80.0%**

Table 14: Blind Human Pairwise Win Ratio of the Second-Round Generation Policy (Round 2) over the First-Round Policy (Round 1) across progressive difficulty intervals.

Comparison	**Overall 0–99**	**Simple 0–49**	**Hard 50–99**	**Sub-Intervals 0–19**	**20–49**	**50–69**	**70–99**
SEER-Gen (R2) vs. R1	**62.25%**	**63.00%**	**61.50%**	**60.00%**	**65.00%**	**58.75%**	**63.33%**

preference probability as $P(x_{\text{pol}} \succ x_{\text{ref}}) = \sigma(r(x_{\text{pol}}) - r(x_{\text{ref}}))$ where the reward $r(x)$ approximates alignment $\log P_\theta(a|x)$. The objective can be interpreted as:

$$\mathcal{J}_{\text{SEER}} \approx \mathbb{E}_{p\sim\pi_\theta}\left[\log P_\theta(a \mid G(p;\phi)) - \beta \log \frac{\pi_\theta(p)}{\pi_{\text{ref}}(p)}\right]. \tag{10}$$

This view is consistent with the unified architecture. Since the evaluator and generator share the model's pretrained representations, the evaluator can provide feedback on whether a reprompt leads to an effective visual realization. A reprompt that produces a poor realization tends to receive a lower preference score and is therefore less favored during RLMT. In this way, SEER encourages the policy to generate descriptions that both express the user intent and better match the generator's capabilities.

A.11 Limitations and Future Work

SEER is currently bounded by the model's latent manifold; it effectively activates dormant capabilities but cannot synthesize concepts entirely absent from pre-training. Future directions include exploring iterative multi-turn reprompting for complex constraints, and extending the framework to jointly optimize the visual generator.