Sketch Then Paint: Hierarchical Reinforcement Learning for Diffusion Multi-Modal Large Language Models

Siqi Luo[1,2,*], **Jianghan Shen**[2,3,*], **Yi Xin**[3,4], **Huayu Zheng**[1], **Haoxing Chen**[3], **Yan Tai**[1], **Yue Li**[5], **Junjun He**[2,4], **Yihao Liu**[2], **Guangtao Zhai**[1], **Yuewen Cao**[2], **Xiaohong Liu**[1,4,†]

[1]Shanghai Jiao Tong University, [2]Shanghai Artificial Intelligence Laboratory, [3]Nanjing University, [4]Shanghai Innovation Institute, [5]Peking University

[*]Equal contribution [†]Corresponding author: xiaohongliu@sjtu.edu.cn

Abstract

Diffusion Multi-Modal Large Language Models (dMLLMs) are powerful for image generation, but optimizing them through reinforcement learning (RL) remains a major challenge. One primary difficulty is that a single image can be generated through many different unmasking sequences, making exact marginalization over generation paths intractable in practice. Additionally, existing methods tend to ignore the hierarchical generation process of dMLLMs, where early tokens define the global layout and later tokens focus on local details. Uniform image-level rewards ignore these distinct token roles. To address these issues, we propose **H**ierarchical **T**oken **GRPO** (**HT-GRPO**), which integrates this hierarchy directly into the policy optimization process. Our approach features a Sketch-Then-Paint training scheme that organizes updates into three distinct stages: global, structure, and refinement. We also use a prompt-conditioned estimator to calculate importance ratios starting from a fully masked state. Furthermore, we introduce a Hierarchical Credit Assignment mechanism that prioritizes key structural tokens to emphasize updates to global structure. Experiments using two popular dMLLM backbones, MMaDA and Lumina-DiMOO, demonstrate that HT-GRPO achieves substantial gains on the GenEval and DPG benchmarks. Evaluations across six automatic metrics show consistent improvements in image quality, aesthetics, and predicted human preference. Project page: https://github.com/Alpha-VLLM/Lumina-DiMOO/tree/main/ht_grpo.

1 Introduction

Recent advancements in text-to-image (T2I) generation, driven by powerful diffusion models [1, 2, 3, 4] and autoregressive (AR) models [5, 6, 7, 8], have achieved significant breakthroughs. This success is largely built upon the scaling of computational resources, model parameters, and training data. Alongside these efforts, the integration of reinforcement learning (RL) [9, 10, 11, 12] has also become an essential component. By effectively aligning generation outputs with human preferences (e.g., prompt adherence, aesthetic), RL significantly improves the performance of T2I models. More recently, Diffusion Multi-Modal Large Language Models (dMLLMs) [13, 14, 15, 16, 17, 18] have emerged as a significant architectural evolution. By performing parallel iterative denoising over discrete tokens, dMLLMs offer a novel approach to visual generation. This architectural shift naturally requires researchers to design RL strategies specifically tailored for dMLLMs. The

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

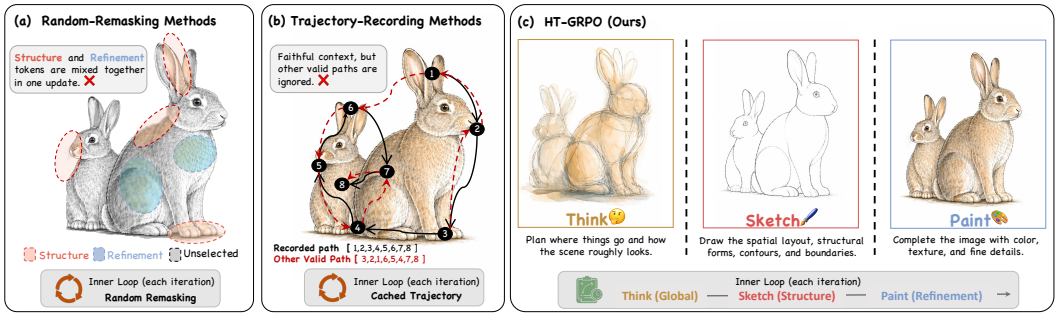


Figure 1: **Comparison of RL inner-loop strategies for dMLLMs.**

generation process of dMLLMs is inherently hierarchical: tokens decoded early shape global layout under high uncertainty, while tokens decoded late refine local details under low uncertainty.

Existing RL approaches generally follow two routes, but both fail to account for this structure: **1) Random remasking methods** (shown in Figure 1(a)) [19, 20, 21] approximate intermediate states by randomly remasking tokens in generated images and evaluating selected tokens under the remaining visible context (*Limitation: future-token contamination*). The retained context can expose future tokens to the selected step. It also arbitrarily mixes structural tokens responsible for global layout with refinement tokens responsible for local details into the same gradient update, preventing the model from optimizing each group at its appropriate stage (*Limitation: stage conflation*). **2) Trajectory recording methods** (shown in Figure 1(b)) [22, 23, 24, 25] build an explicit decision process by recording the denoising trajectory. While this prevents information leakage, the inner-loop updates for each generated image are bound to a single recorded ordering, ignoring the many other valid unmasking orderings (*Limitation: narrow path coverage*). Beyond these specific issues, both routes assign uniform image-level rewards to all tokens, ignoring the fact that different tokens have entirely different impacts on the final image quality (*Shared Limitation: uniform token reward*).

To overcome these limitations, we propose Hierarchical Token GRPO (HT-GRPO), a novel RL method that explicitly encodes the hierarchical structure of dMLLM generation into inner-loop policy optimization. HT-GRPO consists of two core components. **First, we propose a Sketch-Then-Paint staged training framework.** This organizes the inner-loop updates into three consecutive stages (shown in Figure 1(c)): a Global stage that updates all tokens jointly to establish a stable coarse foundation, a Structure stage that focuses exclusively on structural tokens to sharpen global composition, and a Refinement stage that polishes local details only after the structure is settled, separating structural and refinement updates after the joint Global stage. Within each stage, random subset sampling distributes updates across different token subsets without requiring additional denoising trajectories. To support this staged optimization, we further introduce a **prompt-conditioned estimator** that computes token-level importance ratios under a unified fully masked state using only the text prompt as condition. This avoids future-token exposure and intermediate-state caching while using a common prompt-only context for likelihood evaluation. **Second, we introduce Hierarchical Credit Assignment.** This assigns higher credit weights to early structural tokens and lower weights to later refinement tokens, using generation order as a proxy for token role rather than measuring individual causal contributions.

Extensive experiments confirm that HT-GRPO delivers superior text-image alignment on multiple benchmarks. On the MMaDA model, it raises GenEval from 0.56 to 0.83 and DPG from 70.51 to 81.09. When applied to Lumina-DiMOO, these scores further improve to 0.92 and 84.47 respectively. Additionally, HT-GRPO improves performance across six metrics covering image quality, aesthetics, and human preference. These results confirm that hierarchical token optimization is a versatile and effective approach. Overall, our contributions can be summarized as follows:

- We analyze how generation-stage differences motivate hierarchy-aware optimization, highlighting context mismatch and uniform token credit in the compared RL methods.
- We propose **Sketch-Then-Paint** staged training, which divides inner-loop updates into Global, Structure, and Refinement stages by generation order, and uses a prompt-conditioned estimator to prevent future-token contamination without caching trajectories.

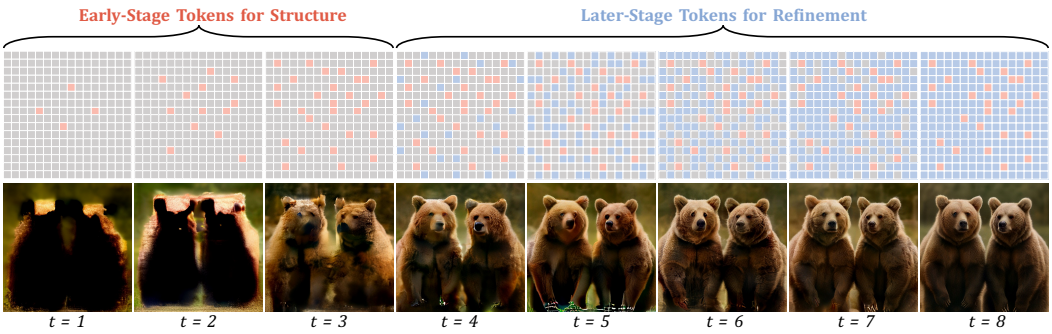


Figure 2: **Visualization of the image generation process in dMLLMs.** Top: early (red, structure) tokens set layout, later (blue, refinement) tokens add detail. Bottom: corresponding generated images.

- We design **Hierarchical Credit Assignment** to allocate generation-order-aware weights to different tokens. This emphasizes structural decisions through a generation-order-based proxy.
- We evaluate HT-GRPO on MMaDA and Lumina-DiMOO, observing gains across GenEval, DPG-Bench, and six automatic preference and quality metrics.

2 Preliminaries

2.1 Image Generation Process of dMLLMs

dMLLMs generate images by iteratively denoising masked visual tokens. An image is represented as a token sequence of length N with text condition c. Starting from a fully masked state $\mathbf{x}^{(T)}$, the model repeatedly unmasks a subset of positions until the final image $\mathbf{x}^{(0)} = (v_1, \ldots, v_N)$ is produced:

$$\mathbf{x}^{(t-1)} \sim \pi_\theta\Big(\cdot \mid \mathbf{x}^{(t)}, c\Big), \qquad t = T, T-1, \ldots, 1. \tag{1}$$

Unlike autoregressive models, which generate tokens in a fixed left-to-right order, dMLLMs can produce the same image through many different unmasking orderings. We track this ordering with the generation-order rank $\rho_{g,i} \in \{1, \ldots, N\}$: rank 1 is the first position unmasked and rank N the last. Tokens unmasked in the same step receive consecutive ranks under an arbitrary fixed order within that step.

The rank $\rho_{g,i}$ reflects a fundamental asymmetry in what the model knows at prediction time. A token with small $\rho_{g,i}$ is unmasked early, when most of the image is still masked. It faces high uncertainty, and its choice largely determines where objects appear and how the scene is organized. A token with large $\rho_{g,i}$ is unmasked late, after most of the image has been revealed. It faces much lower uncertainty and mainly refines color, texture, and local detail. We call tokens with small $\rho_{g,i}$ *structural tokens* and tokens with large $\rho_{g,i}$ *refinement tokens* (Figure 2). Proposition C.1 characterizes the expected entropy decrease under a consistency assumption, rather than guaranteeing monotonicity for every token or trajectory. This rank-induced asymmetry becomes a key challenge when applying RL to dMLLMs, as formalized in Section 2.2.

2.2 Group Relative Policy Optimization

GRPO [26] maximizes a clipped surrogate objective over G rollouts sampled from a behavior policy $\pi_{\theta_{\text{old}}}$, using per-token importance ratios and group-relative advantages; the standard formulation is recalled in Appendix D.

Applying GRPO to dMLLMs. Extending the token-level GRPO objective to dMLLMs rests on two properties of Eq. (1). First, *Markov chain decomposition*: along a fixed unmasking trajectory, the joint probability factorizes step-wise via the chain rule. Second, *within-step conditional independence*: conditioned on the current state $\mathbf{x}^{(t)}$ and text prompt c, the tokens unmasked at step t are sampled independently across positions. Together, these two properties enable a per-token formulation of the importance ratio in dMLLMs, analogous to the autoregressive case (Appendix D).

Assumption 1 (Stage-varying optimization)**.** *Because dMLLMs decode tokens across T stages under progressively changing visual contexts, the per-step optimization objective is inherently non-uniform. We therefore introduce an* optimization support set $\mathcal{M}_g^{(k)}$ *and a* conditioning context $\mathbf{C}_{g,i}^{(k)}$ *to characterize each gradient step k. All methods discussed in this paper are special instantiations of this paradigm. A unified comparison is given in Appendix B.*

Under Assumption 1, the objective at gradient step $k \in \{1, \ldots, K\}$ is:

$$\mathcal{J}^{(k)}(\theta) = \mathbb{E}_{c,\,\{\mathbf{x}_g^{(0)}\}\sim\pi_{\theta_{\text{old}}}}\Bigg[\frac{1}{G}\sum_{g=1}^{G}\frac{1}{|\mathcal{M}_g^{(k)}|}\sum_{i\in\mathcal{M}_g^{(k)}}\min\Big(r_{g,i}^{(k)}(\theta)\,A_g,\ \text{clip}\Big(r_{g,i}^{(k)}(\theta),1-\epsilon,1+\epsilon\Big)\,A_g\Big) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}})\Bigg], \tag{2}$$

where the per-token importance ratio

$$r_{g,i}^{(k)}(\theta) = \frac{\pi_\theta\Big(v_{g,i}\mid \mathbf{C}_{g,i}^{(k)},\,c\Big)}{\pi_{\theta_{\text{old}}}\Big(v_{g,i}\mid \mathbf{C}_{g,i}^{(k)},\,c\Big)} \tag{3}$$

measures the relative likelihood of visual token $v_{g,i}$ under the current versus behavior policy, conditioned on the visual context $\mathbf{C}_{g,i}^{(k)}$ and text context c. The group-relative advantage $A_g = (R_g - \text{mean}(\{R_j\}))/(\text{std}(\{R_j\}) + \delta)$ is a scalar broadcast uniformly to every token in rollout g.

Remark 1. *Each rollout produces one image $\mathbf{x}_g^{(0)}$ via a single unmasking trajectory. The same image could have been generated through exponentially many alternative orderings, each inducing a different conditioning context $\mathbf{C}_{g,i}^{(k)}$ for every token. The ideal objective averages Eq. (2) over all valid trajectories:*

$$\mathcal{J}^*(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{\text{old}}}}[\mathcal{J}_\tau(\theta)]\,, \tag{4}$$

where $\mathcal{J}_\tau$ denotes Eq. (2) instantiated with $\mathcal{M}_g^{(k)} = \{1, \ldots, N\}$ and $\mathbf{C}_{g,i}^{(k)}$ set to the actual state before the denoising step that reveals position i under trajectory τ. This expectation is intractable in practice.

2.3 Limitations of Existing Methods

As noted in Remark 1, $\mathcal{J}^*$ is intractable in practice. Existing methods therefore approximate it by making different choices of $\mathcal{M}_g^{(k)}$ and $\mathbf{C}_{g,i}^{(k)}$. Random remasking methods construct diverse pseudo-trajectories through random masking, while trajectory recording methods condition on the single observed trajectory. Each approximation carries its own limitations, analyzed below. A unified formal comparison is provided in Appendix B.

Random Remasking Methods. MaskGRPO [19], D1 [20], and UniGRPO [21] schedule $\mathcal{M}_g^{(k)}$ by progressively increasing the mask ratio across the K gradient steps. At each step k, the conditioning context $\mathbf{C}_{g,i}^{(k)}$ consists of the randomly retained visible tokens together with the text prompt c. This construction introduces two limitations. First, *future-token contamination*: the randomly retained tokens are not ordered by generation rank. The context $\mathbf{C}_{g,i}^{(k)}$ may therefore include tokens not yet unmasked when position i was originally generated, making it causally inconsistent, as shown in Proposition C.2. Second, *stage conflation*: the progressive mask schedule assigns tokens to $\mathcal{M}_g^{(k)}$ without regard to generation-order rank $\rho_{g,i}$. Structural tokens and refinement tokens are thus mixed in the same gradient update despite their fundamentally different entropy levels, as established in Section 2.1 and formalized in Proposition C.1. On the other hand, the random remasking strategy provides multi-path coverage. By randomly varying the masked tokens across gradient steps, these methods expose the model to multiple partial views of the same generated image, providing broader context diversity than a single fixed trajectory.

Trajectory Recording Methods. TraceRL [22], CJ-GRPO [23], d-TreeRPO [24], and AGRPO [25] set $\mathcal{M}_g^{(k)}$ to all N token positions at every gradient step. The conditioning context $\mathbf{C}_{g,i}^{(k)}$ is the

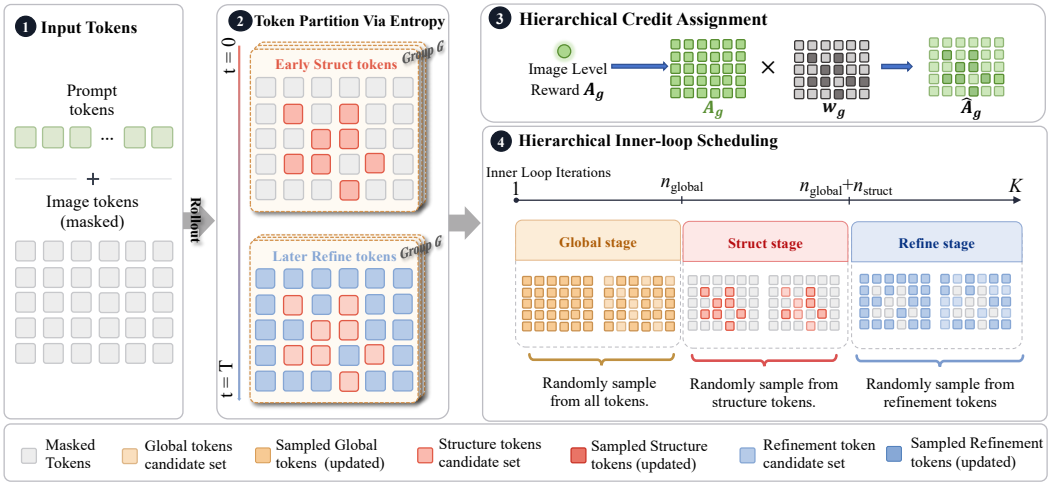


Figure 3: **HT-GRPO framework.** (1) Prompt and masked image tokens are rolled out to produce G sample groups. (2) Tokens are partitioned into structural (early, high-entropy) and refinement (later, low-entropy) sets via generation-order rank. (3) The image-level advantage A_g is reweighted by per-token credits $w_{g,i}$ to form the hierarchical advantage $\tilde{A}_{g,i}$. (4) Inner-loop updates are scheduled into three stages, each randomly sampling from its corresponding token set.

set of tokens already unmasked just before position i was revealed during the rollout trajectory. By grounding the context in the actual generation order, these methods avoid both future-token contamination and stage conflation. They introduce two limitations of their own, however. First, *limited path coverage*: as discussed in Remark 1, the ideal $\mathcal{J}^*$ requires averaging over all valid trajectories. Each rollout records only one ordering, so all alternatives remain unseen and the expectation is approximated by a single sample. Second, computational overhead: trajectory-conditioned ratio evaluation requires $O(T)$ forward passes per inner update. States can be cached or reconstructed from the unmasking-step map. This substantially increases both memory usage and compute cost per training cycle.

Shared Limitation. Neither family accounts for the difference between structural and refinement tokens in the advantage assignment. The scalar A_g is broadcast uniformly to every token regardless of denoising stage, assigning equal credit to structural and refinement tokens despite their vastly different entropy levels.

3 Method

The limitations in Section 2.3 fall into two categories. The first concerns the approximation of $\mathcal{J}^*$: existing choices of $\mathcal{M}_g^{(k)}$ and $\mathbf{C}_{g,i}^{(k)}$ each introduce stage conflation, limited path coverage, or future-token contamination. The second concerns reward attribution: the image-level advantage A_g is broadcast uniformly, ignoring the entropy gap between structural and refinement tokens.

HT-GRPO addresses both categories through two components. Section 3.1 presents Sketch-Then-Paint staged training with a prompt-conditioned estimator. Stage-specific support sets separate Structure and Refinement updates after the joint Global stage. Random subsets vary optimization support, and fixing $\mathbf{C}_{g,i}^{(k)} = \mathbf{C}_\emptyset$ prevents future-token exposure. Section 3.2 replaces the uniform A_g with Hierarchical Credit Assignment.

3.1 Hierarchical Token GRPO

Visual generation is a progressive process from global structure to local detail, encoded in the distribution of $\rho_{g,i}$. As established in Section 2.1, tokens with small $\rho_{g,i}$ are structural tokens facing high uncertainty, while those with large $\rho_{g,i}$ are refinement tokens operating under rich

context. HT-GRPO encodes this ordering into both the optimization schedule and the importance ratio estimator.

3.1.1 Sketch-Then-Paint Staged Training

Under standard GRPO, one policy optimization cycle is:

$$\pi_{\theta_{\text{old}}} \xrightarrow{\text{rollout}} \{\mathbf{x}_g^{(0)}\}_{g=1}^{G} \xrightarrow{\text{optimize}} \theta \xrightarrow{\text{synchronize}} \theta_{\text{old}} \leftarrow \theta. \tag{5}$$

The policy performs K inner-loop gradient updates on the same rollout batch. Although some methods vary which tokens enter each update, none account for the entropy-level distinction between structural and refinement tokens.

We organize the K inner-loop updates into three consecutive stages:

$$\text{Global} \rightarrow \text{Structure} \rightarrow \text{Refinement}. \tag{6}$$

This creates a coarse-to-fine curriculum from high entropy to low entropy. The Global stage first optimizes all tokens jointly to provide a stable starting point. The Structure stage then focuses on high-entropy tokens, and the Refinement stage finally optimizes low-entropy detail after the skeleton has been established. This progression is designed to reduce interference between token groups with different roles.

Let $N_s = \lfloor \alpha N \rfloor$ with $\alpha \in (0, 1)$ denoting the structure fraction. We define the stage-specific token sets as

$$\mathcal{S}_{g,\text{global}} = \{1, \ldots, N\}, \mathcal{S}_{g,\text{structure}} = \{i \mid \rho_{g,i} \leq N_s\}, \mathcal{S}_{g,\text{refinement}} = \{i \mid \rho_{g,i} > N_s\}. \tag{7}$$

Here N_s is a token-count cutoff rather than a denoising-step cutoff. $\mathcal{S}_{g,\text{structure}}$ collects the N_s structural tokens with the smallest $\rho_{g,i}$ values, and $\mathcal{S}_{g,\text{refinement}}$ the remaining $N - N_s$ refinement tokens, following the taxonomy in Section 2.1. This partition depends only on the rollout's own unmasking order and requires no external annotation. Using unmasking rank as an entropy proxy is motivated by Proposition C.1 and the aggregate measurements in Appendix I.3.

We split the total inner-loop budget as $K = n_{\text{global}} + n_{\text{structure}} + n_{\text{refinement}}$. For the k_s-th update inside stage s, where $k_s = 0, \ldots, n_s - 1$, we independently sample a random subset $\mathcal{M}_g^{(k_s)} \subseteq \mathcal{S}_{g,s}$ according to an annealed sampling rate $\gamma_{k_s}^{(s)}$. Only tokens in $\mathcal{M}_g^{(k_s)}$ contribute gradients in that update:

$$\gamma_{k_s}^{(s)} = \gamma_{\min}^{(s)} + \left(\gamma_{\max}^{(s)} - \gamma_{\min}^{(s)}\right) \frac{\max(1, n_s - 1) - k_s}{\max(1, n_s - 1)}. \tag{8}$$

The sampling rate decays linearly from $\gamma_{\max}^{(s)}$ to $\gamma_{\min}^{(s)}$. Early updates cover more tokens, while later updates concentrate the loss on smaller subsets. Sampling different subsets $\mathcal{M}_g^{(k_s)}$ across inner-loop updates ensures that distinct portions of the token space receive gradient signal, rather than committing to a fixed subset throughout optimization. This samples optimization support within each stage. It does not sample alternative trajectory contexts or provide an unbiased estimator of $\mathcal{J}^*$.

3.1.2 Prompt-Conditioned Estimator

Random remasking methods set $\mathbf{C}_{g,i}^{(k)}$ to a randomly retained subset of the generated image, which may expose tokens not yet visible at generation time and causes future-token contamination, as shown in Proposition C.2. Trajectory recording methods use the actual denoising states and require up to T forward passes per inner update, with states cached or reconstructed.

We instead use the fully masked initial state $\mathbf{x}^{(T)}$ as a unified conditioning context, denoted $\mathbf{C}_\emptyset$ to emphasize that all N positions are masked, and define the prompt-conditioned importance ratio as

$$r_{g,i}(\theta) = \frac{\pi_\theta(v_{g,i} \mid \mathbf{C}_\emptyset, c)}{\pi_{\theta_{\text{old}}}(v_{g,i} \mid \mathbf{C}_\emptyset, c)}. \tag{9}$$

Under a fully masked input, dMLLMs predict all N token distributions in a single forward pass, so all probabilities are obtained without caching any intermediate states. Using $\mathbf{C}_\emptyset$ eliminates future-token contamination by construction, preserves predictive entropy for refinement tokens as shown in Proposition C.3, and avoids caching intermediate denoising states. The ratio is a surrogate evaluated under a different context, not the exact trajectory-consistent importance ratio. Appendix I.2 quantifies this distinction.

Table 1: **Performance comparison on GenEval** across various dMLLMs and RL settings.

Methods	Single ↑	Two ↑	Counting ↑	Colors ↑	Position ↑	Attribute ↑	Overall ↑
Text-to-Image Models							
SDXL [27]	0.98	0.74	0.39	0.85	0.15	0.23	0.55
DALL-E 3 [28]	0.96	0.87	0.47	0.83	0.43	0.45	0.67
Janus-Pro [29]	0.99	0.89	0.59	0.90	0.79	0.66	0.80
FLUX.1-dev [30]	0.98	0.93	0.75	**0.93**	0.68	0.65	0.82
dMLLMs (wo / w RL)							
MMaDA [13]	0.96	0.60	0.45	0.81	0.14	0.25	0.56
+ UniGRPO [13]	0.99	0.76	0.61	0.84	0.20	0.37	0.63
+ MaskGRPO [19]	0.99	0.85	0.66	0.89	**0.73**	**0.69**	0.80
+ TraceGRPO [22]	1.00	0.87	0.72	**0.92**	0.64	0.62	0.79
+ HT-GRPO (Ours)	**1.00**	**0.92**	**0.77**	0.90	**0.73**	0.66	**0.83**
Lumina-DiMOO [14]	0.99	0.94	0.70	0.85	0.81	0.71	0.83
+ MaskGRPO [19]	**1.00**	0.90	0.88	0.89	0.85	**0.77**	0.88
+ TraceGRPO [22]	**1.00**	0.90	0.85	0.90	0.88	0.73	0.87
+ HT-GRPO (Ours)	**1.00**	**0.95**	**0.94**	**0.93**	**0.93**	0.75	**0.92**

3.2 Hierarchical Credit Assignment

Both families of existing methods uniformly broadcast the image-level advantage A_g to all tokens, assigning equal credit to structural and refinement tokens despite their fundamentally different roles. We instead assign token-level credit weights according to generation order:

$$\tilde{A}_{g,i} = A_g \cdot w_{g,i}, \qquad w_{g,i} = \begin{cases} \lambda_{\mathrm{s}}, & i \in \mathcal{S}_{g,\mathrm{structure}}, \\ \lambda_{\mathrm{r}}, & i \in \mathcal{S}_{g,\mathrm{refinement}}, \end{cases} \tag{10}$$

where $\lambda_{\mathrm{s}} > 1 > \lambda_{\mathrm{r}} \geq 0$. Structural tokens receive weights above the uniform baseline to amplify updates on global composition and layout, while refinement tokens receive smaller weights to attenuate updates on local detail. All three training stages share the same weighting scheme. The only difference across stages is which active token set $\mathcal{S}_{g,s}$ is selected.

The HT-GRPO per-step objective is Eq. (2) with A_g replaced by $\tilde{A}_{g,i}$:

$$\mathcal{J}^{(k)}(\theta) = \mathbb{E}_{c,\,\{\mathbf{x}_g^{(0)}\}\sim\pi_{\theta_{\mathrm{old}}}}\Bigg[\frac{1}{G}\sum_{g=1}^{G}\frac{1}{|\mathcal{M}_g^{(k)}|}\sum_{i\in\mathcal{M}_g^{(k)}}\min\Big(r_{g,i}(\theta)\,\tilde{A}_{g,i},\ \mathrm{clip}(r_{g,i}(\theta),1-\epsilon,1+\epsilon)\,\tilde{A}_{g,i}\Big) - \beta\,\mathbb{D}_{\mathrm{KL}}(\pi_\theta\,\|\,\pi_{\mathrm{ref}})\Bigg], \tag{11}$$

where the K updates are ordered as Global → Structure → Refinement, consistent with Appendix B.

4 Experiments

4.1 Experimental Setup

Models. We validate our proposed HT-GRPO on two popular open-source pre-trained dMLLMs: MMaDA [13] and Lumina-DiMOO [14]. Since other notable dMLLMs, such as LaViDa-O [17], are not open-source, we cannot include them in our experiments. Additionally, models like LLaDA-o [18] are not strictly dMLLMs because they rely on continuous diffusion for image generation, making them unsuitable as a base model.

Baselines. Our primary baselines include specialized T2I models (e.g., SDXL [27], FLUX.1-dev [30], DALL-E 3 [28], Janus-Pro [29]), as well as the base versions of MMaDA and Lumina-DiMOO. Furthermore, we compare various RL training strategies. Specifically, we evaluate MaskGRPO [19] on the same dMLLM backbones to highlight the advantages of HT-GRPO. We additionally compare

Table 2: **Performance comparison on DPG-Bench** across various dMLLMs and RL settings.

Model	Global ↑	Entity ↑	Attribute ↑	Relation ↑	Other ↑	Overall ↑
Text-to-Image Models						
SDXL [27]	83.27	82.43	80.91	86.76	80.41	74.65
DALL-E 3 [28]	90.97	89.61	88.39	90.58	89.83	83.50
Janus-Pro [29]	86.90	88.90	89.40	89.32	89.48	84.19
FLUX.1-dev [30]	74.35	90.00	88.96	90.87	88.33	83.84
dMLLMs (wo / w RL)						
MMaDA [13]	77.52	77.52	77.67	81.56	63.13	70.51
+ MaskGRPO [19]	80.92	79.52	85.41	83.05	69.12	75.81
+ TraceGRPO [22]	82.34	78.92	81.29	82.08	70.21	74.37
+ HT-GRPO (Ours)	**89.76**	**87.62**	**86.58**	**88.86**	**81.83**	**81.09**
Lumina-DiMOO [14]	81.17	89.96	88.10	89.64	79.95	81.86
+ MaskGRPO [19]	82.65	89.50	88.33	88.60	**88.24**	82.24
+ TraceGRPO [22]	84.89	89.65	88.23	90.24	84.28	82.62
+ HT-GRPO (Ours)	**88.04**	**91.22**	**88.98**	**92.56**	83.05	**84.47**

against TraceGRPO, a trajectory-aware baseline that constructs causally consistent contexts from the recorded unmasking order. Following the implementation of TraceRL [22], we refer to this method as TraceGRPO. We also test UniGRPO to establish the performance bound for single-stage RL.

Benchmarks & Metrics. To evaluate T2I generation performance, we employ two related benchmarks: GenEval [31] and DPG-Bench [32]. GenEval, consisting of 553 text prompts, is the most widely used benchmark to assess object-centric generation. In contrast, DPG-Bench includes 1,065 dense and complex prompts to evaluate how well the generated images align with the text. Beyond benchmark performance, we also measure the human preference, visual quality, and aesthetics of the generated images using ImageReward [33], DeQA [34], HPSv3 [35], and UniPercept [36].

Implementation Details. HT-GRPO is implemented on top of the MaskGRPO [19] codebase, inheriting its optimizer, reward, and RL loop. Specifically, we generate $G = 9$ rollouts per prompt. The final reward score is a combination of HPSv3 [35], CLIP score [37], and UniReward [38], maintaining the original KL penalty coefficient and fixing the classifier-free guidance at 3.5. HT-GRPO-specific settings are: structure ratio $\alpha = 0.3$, three-stage budget $n_{\text{global}} : n_{\text{structure}} : n_{\text{refinement}} = 2 : 4 : 2$ ($K = 8$), token-level credit weights $\lambda_{\text{s}} = 1.5$ and $\lambda_{\text{r}} = 0.5$, and linear-decay annealing ($\gamma_{\max} = 1.0$, $\gamma_{\min} = 0.5$, down mode). The main image-generation experiments run on 8× A100-80G GPUs. The additional efficiency profiling uses 2× H200 GPUs, as specified below. More details are provided in Appendix F.

Hyperparameter selection. We use the same default settings on both backbones without exhaustive per-backbone or per-benchmark searches. The boundary $\alpha = 0.3$ reserves early tokens for structure, the 2:4:2 budget allocates half the updates to structure, and $(\lambda_{\text{s}}, \lambda_{\text{r}}) = (1.5, 0.5)$ prioritizes these tokens. These are design-guided defaults, not theoretically optimal constants. Appendix I.4 reports local sensitivity results.

4.2 Main Results

GenEval. Table 1 shows that HT-GRPO improves MMaDA from 0.56 to 0.83, exceeding UniGRPO (0.63), MaskGRPO (0.80), and TraceGRPO (0.79). On Lumina-DiMOO, it improves the score from 0.83 to 0.92, compared with 0.88 for MaskGRPO and 0.87 for TraceGRPO. The gains on both backbones support the effectiveness of hierarchical token optimization. Among the compared specialized T2I models, FLUX.1-dev scores 0.82, while Lumina-DiMOO with HT-GRPO achieves the highest score at 0.92.

DPG-Bench. Table 2 shows that HT-GRPO raises MMaDA from 70.51 to 81.09 and Lumina-DiMOO from 81.86 to 84.47. On Lumina-DiMOO, MaskGRPO and TraceGRPO reach 82.24 and 82.62, respectively. HT-GRPO also exceeds DALL-E 3 (83.50) and FLUX.1-dev (83.84) on this benchmark. For the six automatic preference, quality, and aesthetics metrics in Table 3, HT-GRPO leads on five

Table 3: **Human preference, image quality, and aesthetics comparison on DPG-Bench** across various dMLLMs and RL settings.

Methods	DeQA ↑	ImgReward ↑	HPSv3 ↑	UniPercept		
				Aesthetics ↑	Quality ↑	Structure ↑
MMaDA [13]	3.91	0.64	8.81	58.83	65.29	47.46
+ MaskGRPO [19]	4.05	0.83	9.40	59.02	67.07	48.15
+ TraceGRPO [22]	**4.08**	0.84	9.56	58.78	66.87	49.04
+ HT-GRPO (Ours)	4.07	**0.86**	**10.46**	**59.29**	**67.91**	**49.45**
Lumina-DiMOO [14]	4.03	0.84	11.60	56.66	66.27	43.12
+ MaskGRPO [19]	4.08	0.90	12.00	56.77	66.13	43.55
+ TraceGRPO [22]	4.10	0.88	11.28	57.68	67.21	43.89
+ HT-GRPO (Ours)	**4.12**	**0.97**	**12.46**	**58.39**	**67.97**	**44.97**

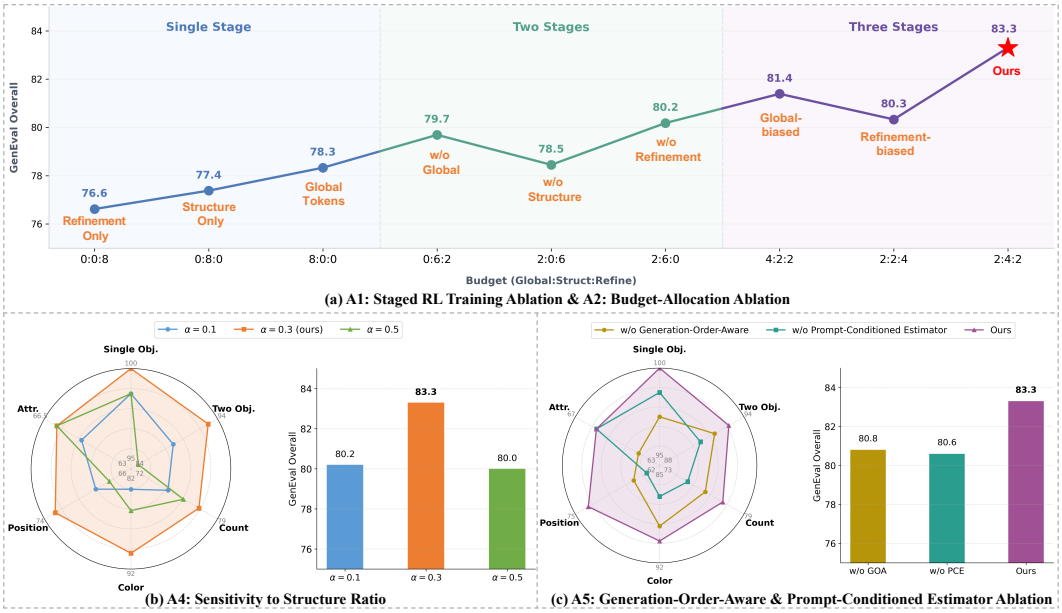


Figure 4: **(a) Stage organization and budget allocation:** The full Global→Structure→Refinement schedule with a structure-biased 2:4:2 budget consistently outperforms single-stage and two-stage variants. **(b) Sensitivity to structure ratio α:** Performance peaks at $\alpha = 0.3$ and degrades when the boundary is too narrow or too broad. **(c) Component analysis:** Hierarchical Credit Assignment and Prompt-Conditioned Estimator each provide independent gains, validating both design choices.

for MMaDA and all six for Lumina-DiMOO among the compared variants. The exception is MMaDA DeQA, where TraceGRPO scores 4.08 versus 4.07 for HT-GRPO. These results support improved prompt alignment and perceptual scores, while showing that gains can vary by metric.

4.3 Efficiency and Estimator Diagnostics

As shown in Figure 5, under matched MMaDA profiling on 2×H200 GPUs with ZeRO-2, global batch size 72, $T = 12$, and $K = 8$, HT-GRPO uses 173.7 TFLOPs and 0.17 seconds per update, versus 1490.8 TFLOPs and 1.54 seconds for TraceGRPO. Peak memory is 18.9 versus 31.3 GB. The shared rollout costs 1484.5 TFLOPs and 8.36 seconds per cycle, so these update-phase comparisons do not represent end-to-end training speedups. HT-GRPO has nearly the same measured cost as MaskGRPO (Appendix I.1).

After the shared rollout, TraceGRPO evaluates up to T trajectory-conditioned states per inner update, whereas HT-GRPO evaluates one fully masked context. The measured difference is therefore in post-rollout optimization, with the same T-step generation procedure. TraceGRPO requires 8.58 times the update FLOPs and 9.06 times the update latency of HT-GRPO in this matched profile. The peak-memory reduction is 12.4 GB, while memory use remains nearly identical to MaskGRPO.

The prompt-conditioned ratio has log-ratio MAE 0.0194–0.0254 across noninitial generation phases, but sign agreement with the trajectory-consistent ratio is only 0.499–0.505. It is therefore not a faithful tokenwise substitute, and near-zero mean error does not establish unbiasedness (Appendix I.2).

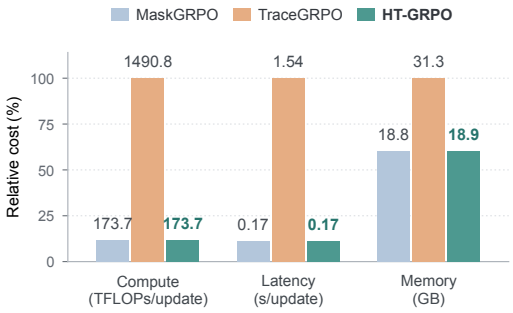


Figure 5: **MMaDA training cost.** Heights are relative to TraceGRPO (100%); labels show absolute values. Compute and latency exclude rollouts.

4.4 Ablation Study

Figure 4 and Table 5 (shown in Appendix E) summarize five ablation studies on MMaDA ($K = 8$, $\alpha = 0.3$). **(a) Stage organization and budget:** The full Global→Structure→Refinement schedule with a 2:4:2 budget reaches 83.3 overall, outperforming all single-stage and two-stage variants (76.6–80.2); the coarse-to-fine ordering itself is critical. **(b) Structure ratio:** Performance peaks at $\alpha = 0.3$ and degrades for narrower ($\alpha = 0.1$, 80.2) and broader ($\alpha = 0.5$) boundaries, confirming that a well-calibrated hierarchy is essential. **(c) Component analysis:** Removing hierarchical credit weighting ($\lambda_s = \lambda_r = 1$) drops the score to 80.8, and replacing $\mathbf{C}_\emptyset$ with revealed structural contexts reduces it to 80.6, validating both designs independently. The linear-decay annealing schedule (Table 5, 83.31) outperforms static and ascending baselines, confirming the value of broad initial coverage for stable gradient directions.

Further robustness checks support these defaults within the evaluated settings. Structure-prioritized credit pairs yield GenEval 83.3–83.6, compared with 80.8 for uniform credit and 77.9 for reversed credit (Appendix I.4). Entropy measurements on 756 trajectories per backbone show a decline from the unmask-ratio window $[0.2, 0.4]$ onward, with 95% cluster-bootstrap intervals below zero (Appendix I.3). These results support a practical transition range and local weight robustness, rather than a universal boundary or strict per-trajectory monotonicity. Full per-ablation details are provided in Appendix E.

5 Conclusion

We revisit the application of GRPO to dMLLMs and identify two key challenges: multiple valid unmasking orderings make importance-ratio estimation difficult, while existing methods uniformly assign image-level rewards to tokens with heterogeneous roles. To address these issues, we propose HT-GRPO, which uses generation-order rank to characterize token uncertainty and role, organizes inner-loop updates into Global, Structure, and Refinement stages, estimates importance ratios under a fully masked prompt-conditioned context, and performs generation-order-aware credit assignment. Experiments on MMaDA and Lumina-DiMOO show consistent improvements across GenEval, DPG-Bench, and automatic human-preference and visual-quality metrics, supporting hierarchical token optimization.

HT-GRPO uses a fixed structure ratio and fixed credit weights, which may not adapt optimally to prompt complexity or rollout uncertainty. Its prompt-conditioned ratio is a surrogate with token-level directional mismatch relative to trajectory-consistent ratios, and the analysis does not establish an unbiased policy gradient. Evidence beyond image-generating dMLLMs is preliminary (Appendix I.5). Adaptive token grouping and dynamic credit assignment are promising directions for future work.

Acknowledgments and Disclosure of Funding

This work was sponsored by the Fundamental Research Funds for the Central Universities (project number YG2024QNA22) and supported by the National Natural Science Foundation of China under Grant 62572317. The authors declare no competing interests or relevant financial relationships that could be perceived as influencing this work.

References

[1] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[2] Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Jiakang Yuan, Xinyue Li, Dongyang Liu, et al. Lumina-image 2.0: A unified and efficient image generative framework. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2025.

[3] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[4] Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

[5] 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*, 2024.

[6] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, et al. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[7] Yi Xin, Juncheng Yan, Qi Qin, Zhen Li, Dongyang Liu, Shicheng Li, Victor Shea-Jay Huang, Yupeng Zhou, Renrui Zhang, Le Zhuo, et al. Lumina-mgpt 2.0: Stand-alone autoregressive image modeling. *arXiv preprint arXiv:2507.17801*, 2025.

[8] Dongyang Liu, Yi Xin, Shitian Zhao, Le Zhuo, Weifeng Lin, Xinyue Li, Qi Qin, Guangtao Zhai, Xiaohong Liu, Hongsheng Li, et al. Lumina-mgpt: Flexible photorealistic autoregressive text-to-image generation. *International Journal of Computer Vision (IJCV)*, 2026.

[9] 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.

[10] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[11] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

[12] Shihao Yuan, Yahui Liu, Yang Yue, Jingyuan Zhang, Wangmeng Zuo, Qi Wang, Fuzheng Zhang, and Guorui Zhou. Ar-grpo: Training autoregressive image generation models via reinforcement learning. *arXiv preprint arXiv:2508.06924*, 2025.

[13] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[14] Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

[15] Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song, Yunhai Tong, Xiangtai Li, Xuelong Li, and Shuicheng Yan. Beyond text-to-image: Liberating generation with a unified discrete diffusion model. In *International Conference on Learning Representations*, 2026.

[16] Tiwei Bie, Haoxing Chen, Tieyuan Chen, Zhenglin Cheng, Long Cui, Kai Gan, Zhicheng Huang, Zhenzhong Lan, Haoquan Li, Jianguo Li, Tao Lin, Qi Qin, Hongjun Wang, Xiaomei Wang, Haoyuan Wu, Yi Xin, and Junbo Zhao. Llada2.0-uni: Unifying multimodal understanding and generation with diffusion large language model. *arXiv preprint arXiv:2604.20796*, 2026.

[17] Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation. *arXiv preprint arXiv:2509.19244*, 2025.

[18] Zebin You, Xiaolu Zhang, Jun Zhou, Chongxuan Li, and Ji-Rong Wen. Llada-o: An effective and length-adaptive omni diffusion model. *arXiv preprint arXiv:2603.01068*, 2026.

[19] Tianren Ma, Mu Zhang, Yibing Wang, and Qixiang Ye. Consolidating reinforcement learning for multimodal discrete diffusion models. *arXiv preprint arXiv:2510.02880*, 2025.

[20] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

[21] Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, and Wanli Ouyang. UniGRPO: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026.

[22] Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025.

[23] Jingyi Yang, Guanxu Chen, Xuhao Hu, and Jing Shao. Taming masked diffusion language models via consistency trajectory reinforcement learning with fewer decoding step. *arXiv preprint arXiv:2509.23924*, 2025.

[24] Leyi Pan, Shuchang Tao, Yunpeng Zhai, Zheyu Fu, Liancheng Fang, Minghua He, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. d-treerpo: Towards more reliable policy optimization for diffusion language models. *arXiv preprint arXiv:2512.09675*, 2025.

[25] Anthony Zhan. Simple policy gradients for reasoning with diffusion language models. *arXiv preprint arXiv:2510.04019*, 2025.

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

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

[28] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Tian, et al. Improving image generation with better captions. *Technical Report, OpenAI*, 2023.

[29] 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*, 2025.

[30] Black Forest Labs. FLUX.1, 2024. Official model release.

[31] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 76341–76366, 2023.

[32] 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. Introduces DPG-Bench.

[33] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[34] Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[35] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. *arXiv preprint arXiv:2508.03789*, 2025.

[36] Shuo Cao, Jiayang Li, Xiaohui Li, Yuandong Pu, Kaiwen Zhu, Yuanting Gao, Siqi Luo, Yi Xin, Qi Qin, Yu Zhou, Xiangyu Chen, Wenlong Zhang, Bin Fu, Yu Qiao, and Yihao Liu. Unipercept: Towards unified perceptual-level image understanding across aesthetics, quality, structure, and texture. *arXiv preprint arXiv:2512.21675*, 2025.

[37] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. https://github.com/taited/clip-score, March 2023. Version 0.2.1.

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

[39] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[40] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[41] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[42] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[43] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. LLaDA 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

[44] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. LLaDA-V: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

[45] Ye Tian, Ling Yang, Jiongfan Yang, Anran Wang, Yu Tian, Jiani Zheng, Haochen Wang, Zhiyang Teng, Zhuochen Wang, Yinjie Wang, Yunhai Tong, Mengdi Wang, and Xiangtai Li. Mmada-parallel: Multimodal large diffusion language models for thinking-aware editing and generation. *arXiv preprint arXiv:2511.09611*, 2025.

[46] Yi Xin, Siqi Luo, Qi Qin, Haoxing Chen, Kaiwen Zhu, Zhiwei Zhang, Yangfan He, Rongchao Zhang, Jinbin Bai, Shuo Cao, et al. dmllm-tts: Self-verified and efficient test-time scaling for diffusion multi-modal large language models. *arXiv preprint arXiv:2512.19433*, 2025.

[47] Kaiwen Zhu, Quansheng Zeng, Yuandong Pu, Shuo Cao, Xiaohui Li, Yi Xin, Qi Qin, Jiayang Li, Yu Qiao, Jinjin Gu, and Yihao Liu. Accelerating masked image generation by learning latent controlled dynamics. *arXiv preprint arXiv:2602.23996*, 2025.

[48] Shengjun Zhang, Zhang Zhang, Chensheng Dai, and Yueqi Duan. E-GRPO: High entropy steps drive effective reinforcement learning for flow models. *arXiv preprint arXiv:2601.00423*, 2026.

[49] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Supplementary Material

A Related Work

A.1 Diffusion Multi-Modal Large Language Models

Discrete diffusion modeling is rapidly emerging as a highly promising paradigm, showing great potential to replace the traditional autoregressive (AR) modeling. Early discrete diffusion methods [39, 40] built the foundation for this shift by establishing the core principles of token-level denoising. Building on these methods, diffusion large language models (dLLMs) [41, 42, 43] show that text generation can be treated as a parallel and iterative denoising process. Expanding this framework to multi-modal domains, LLaDA-V [44] used visual instruction tuning to show that diffusion models can effectively combine visual and textual data. Driven by this progress, recent studies focus on developing unified diffusion multi-modal large language models (dMLLMs) [13, 15, 14, 16]. These models further extend the capabilities of image generation. This success also inspires new research on downstream applications for dMLLMs. Key areas include reinforcement learning algorithm design [19, 13], image editing [45], test-time scaling [46], and sampling acceleration [47].

A.2 Reinforcement Learning Alignment for dMLLMs

Applying Reinforcement Learning (RL) to discrete diffusion models introduces unique challenges, primarily because dynamic masking breaks the standard policy gradient assumption. To bypass this, existing works generally diverge into two directions. Random remasking methods [19, 20, 21] approximate intermediate states by randomly masking tokens. While maintaining the efficiency of a single forward pass, these approximated contexts often deviate from the actual denoising trajectory. Conversely, trajectory recording methods [22, 23, 24, 25] cache the entire denoising process to build a precise Markov Decision Process (MDP). However, this rigorousness comes at a heavy price, as the computational and memory costs scale linearly with the number of denoising steps. Crucially, both paradigms share a fundamental flaw: they uniformly assign the final reward to all tokens, ignoring prior insights [48] that early, high-entropy steps matter more for generation quality.

B A Unified Theoretical Framework for dMLLM RL Methods

This section formalizes the two existing approaches described in Section 2, random remasking methods and trajectory recording methods, together with HT-GRPO. We use g, i, k, and c for rollout index, token position, inner-loop update index, and text prompt condition respectively. All other symbols follow the notation in the main text.

As established in Section 2.3, the surrogate ideal objective $\mathcal{J}^*$ (Eq. (4)) requires averaging over all valid denoising trajectories $\tau \in \mathcal{T}$, which is combinatorially intractable. All three paradigms therefore share the same clipped surrogate objective and differ only in three design choices: the conditioning context $\mathbf{C}_{g,i}^{(k)}$, the optimization support set $\mathcal{M}_g^{(k)}$, and the token-level weight $w_{g,i}$. Setting these to the choices in Section 3 recovers HT-GRPO exactly.

$$\mathcal{L}^{(k)}(\theta) = \frac{1}{G}\sum_{g=1}^{G}\frac{1}{|\mathcal{M}_g^{(k)}|}\sum_{i\in\mathcal{M}_g^{(k)}}\min\Big(r_{g,i}^{(k)}(\theta)\,\tilde{A}_{g,i},\ \mathrm{clip}\Big(r_{g,i}^{(k)}(\theta),1-\epsilon,1+\epsilon\Big)\,\tilde{A}_{g,i}\Big), \tag{12}$$

where

$$r_{g,i}^{(k)}(\theta) = \frac{\pi_\theta\Big(v_{g,i}\mid \mathbf{C}_{g,i}^{(k)},c\Big)}{\pi_{\theta_{\text{old}}}\Big(v_{g,i}\mid \mathbf{C}_{g,i}^{(k)},c\Big)}. \tag{13}$$

Here, $k \in \{1,\dots,K\}$ indexes the inner-loop gradient step, and the same rollout batch is reused for all K updates. $v_{g,i}$ is the token value generated at position i in rollout g. $\mathcal{M}_g^{(k)}$ is the optimization support set, the set of token positions that contribute gradients in the k-th inner-loop update. The conditioning context $\mathbf{C}_{g,i}^{(k)}$ is the partially observed token configuration used to evaluate the importance ratio for

position i in update k. The token-level weighted advantage $\tilde{A}_{g,i} = w_{g,i} A_g$ scales the group-relative advantage A_g by a per-token credit weight $w_{g,i}$, where

$$A_g = \frac{R_g - \text{mean}(\{R_j\})}{\text{std}(\{R_j\}) + \delta_m}. \tag{14}$$

The three methods differ only in how they instantiate these three quantities.

B.1 Random Remasking Methods

Random remasking methods, including MaskGRPO [19], D1 [20], and UniGRPO [21], retain only the final image $\mathbf{x}_g$ and construct synthetic contexts by independently remasking each position with probability γ_k:

$$\begin{aligned} \mathcal{M}_g^{(k)} &= \{i : i \text{ is remasked with prob. } \gamma_k\} && \text{(support set)}, \\ \mathbf{C}_{g,i}^{(k)} &= \text{Mask}(\mathbf{x}_g, \mathcal{M}_g^{(k)}) && \text{(mask } \mathcal{M}_g^{(k)}\text{; retain token values elsewhere)}, \\ w_{g,i} &= 1 && \text{(uniform weight)}. \end{aligned} \tag{15}$$

Stage conflation. Because remasking is independent of the generation-order rank $\rho_{g,i}$, the support set $\mathcal{M}_g^{(k)}$ mixes structural tokens (small $\rho_{g,i}$) and refinement tokens (large $\rho_{g,i}$) in the same update with equal probability. By Proposition C.1, these two groups carry fundamentally different levels of uncertainty, so their gradients operate at very different scales within a single update step.

Future-token contamination. The context $\mathbf{C}_{g,i}^{(k)}$ retains all positions not in $\mathcal{M}_g^{(k)}$. For position i, define the future token set as $\mathcal{F}_i^{(g)} = \{j : \rho_{g,j} > \rho_{g,i}\}$. Since remasking ignores generation order, each position in $\mathcal{F}_i^{(g)}$ is retained in $\mathbf{C}_{g,i}^{(k)}$ with probability $1 - \gamma_k$, potentially exposing tokens that had not yet been generated when position i was predicted. Proposition C.2 shows this occurs with strictly positive probability for any non-final position.

Multi-path coverage (partial). Different random masks induce diverse conditioning contexts, exploring a broader range of token-context combinations than a single fixed trajectory. Random remasking is agnostic to generation order, however, so the resulting contexts are not guaranteed to correspond to any valid denoising trajectory. Because the context may include future tokens, as shown in the contamination analysis above, the diverse paths explored do not respect the causal ordering of the original generation process. The approximation of $\mathcal{J}^*$ is therefore partial: broader in context diversity than a single trajectory, but biased away from the causal structure that $\mathcal{J}^*$ requires.

Uniform token reward (unresolved). All positions receive the same credit weight $w_{g,i} = 1$ regardless of their generation-order rank. Structural and refinement tokens carry fundamentally different entropy levels and play different roles in image generation, as established in Section 2.3. Equal credit assignment ignores this asymmetry entirely.

B.2 Trajectory Recording Methods

Trajectory recording methods, including TraceRL [22], CJ-GRPO [23], d-TreeRPO [24], and AGRPO [25], record the denoising order and evaluate each token under its trajectory-consistent conditioning context:

$$\begin{aligned} \mathcal{M}_g^{(k)} &= \{1, \ldots, N\} && \text{(all } N \text{ tokens per update)}, \\ \mathbf{C}_{g,i}^{(k)} &= \mathbf{x}_g^{(\prec i)} && \text{(actual state before the step revealing } i\text{)}, \\ w_{g,i} &= 1 && \text{(uniform weight)}. \end{aligned} \tag{16}$$

Stage conflation (avoided). Unlike random remasking methods, which apply the same synthetic context to all tokens in $\mathcal{M}_g^{(k)}$ regardless of generation rank, trajectory recording methods set $\mathcal{M}_g^{(k)} = \{1, \ldots, N\}$ and assign each token its own trajectory-consistent context $\mathbf{C}_{g,i}^{(k)} = \mathbf{x}_g^{(\prec i)}$. Positions unmasked in the same step share the actual pre-step context, regardless of their within-step rank. Early structural tokens therefore receive sparse contexts, while later refinement tokens generally receive richer contexts. Each importance ratio is computed under a context that matches the entropy

level of the corresponding token, avoiding the uniform-context mismatch that characterizes stage conflation.

Future-token contamination (avoided). The trajectory-consistent context $\mathbf{C}_{g,i}^{(k)} = \mathbf{x}_g^{(\prec i)}$ reveals only tokens from earlier denoising steps. All positions in the future token set $\mathcal{F}_i^{(g)} = \{j : \rho_{g,j} > \rho_{g,i}\}$ are masked by construction, so future-token contamination cannot occur.

Limited path coverage. Both $\mathcal{M}_g^{(k)}$ and $\mathbf{C}_{g,i}^{(k)}$ are derived from a single rollout trajectory, so each update evaluates only one element of the expectation in Eq. (4). The many other valid trajectories $\tau \in \mathcal{T}$ that could produce the same image are never seen, restricting the diversity of generation paths the model learns from.

Uniform token reward (unresolved). All positions receive the same credit weight $w_{g,i} = 1$. Although the trajectory-consistent context $\mathbf{x}_g^{(\prec i)}$ naturally differentiates the conditioning richness of structural versus refinement tokens, the advantage broadcast to each token remains uniform. There is no explicit mechanism to amplify the learning signal for high-uncertainty structural tokens or attenuate it for near-deterministic refinement tokens.

B.3 HT-GRPO

HT-GRPO retains the final sample together with the generation-order rank $\rho_{g,i} \in \{1, \ldots, N\}$, which records the unmasking step at which position i was revealed in rollout g, with $\rho_{g,i} = 1$ for the first token and $\rho_{g,i} = N$ for the last. Positions unmasked in the same denoising step share the same context, but their predictive entropies can differ despite consecutive rank values. HT-GRPO partitions tokens into global, structural, and refinement groups:

$$\begin{aligned} \mathcal{S}_{g,\text{global}} &= \{1, \ldots, N\}, \\ \mathcal{S}_{g,\text{structure}} &= \{i : \rho_{g,i} \le N_s\}, \\ \mathcal{S}_{g,\text{refinement}} &= \{i : \rho_{g,i} > N_s\}, \end{aligned} \tag{17}$$

where $N_s = \lfloor \alpha N \rfloor$ and $K = n_{\text{global}} + n_{\text{structure}} + n_{\text{refinement}}$ is the total number of inner-loop updates. The K steps follow a fixed Global $\rightarrow$ Structure $\rightarrow$ Refinement schedule, and the support set at step k is a random subset drawn from the active stage:

$$\mathcal{M}_g^{(k)} \subseteq \begin{cases} \mathcal{S}_{g,\text{global}}, & 1 \le k \le n_{\text{global}}, \\ \mathcal{S}_{g,\text{structure}}, & n_{\text{global}} < k \le n_{\text{global}} + n_{\text{structure}}, \\ \mathcal{S}_{g,\text{refinement}}, & n_{\text{global}} + n_{\text{structure}} < k \le K. \end{cases} \tag{18}$$

All tokens share the same fully masked conditioning context, defined as the state in which all N image positions remain masked:

$$\mathbf{C}_{g,i}^{(k)} \equiv \mathbf{C}_{\emptyset}. \tag{19}$$

Token weights are generation-order-aware rather than uniform:

$$\begin{aligned} \tilde{A}_{g,i} &= w_{g,i} A_g, \\ w_{g,i} &= \begin{cases} \lambda_{\text{s}}, & i \in \mathcal{S}_{g,\text{structure}}, \\ \lambda_{\text{r}}, & i \in \mathcal{S}_{g,\text{refinement}}. \end{cases} \end{aligned} \tag{20}$$

Stage conflation (resolved). At each inner-loop step k, the support set $\mathcal{M}_g^{(k)}$ is drawn exclusively from the token set of the active stage. The Global $\rightarrow$ Structure $\rightarrow$ Refinement schedule partitions the K updates by generation rank, so the Structure and Refinement stages use disjoint token groups after a Global stage that intentionally updates both groups jointly. The entropy gap between these two groups is established in Proposition C.1. This resolves the stage conflation identified in Section 2.3.

Future-token contamination (eliminated). Setting $\mathbf{C}_{g,i}^{(k)} \equiv \mathbf{C}_{\emptyset}$ masks all N positions uniformly for every token and every update, making future-token exposure impossible by construction. Contamination is eliminated without requiring trajectory storage, as formalized in Proposition C.2.

Limited path coverage (alleviated). Within each stage, $\mathcal{M}_g^{(k)}$ is re-sampled independently at every inner-loop step according to an annealed sampling rate $\gamma_{k_s}^{(s)}$ (Appendix F.1). Different token subsets

receive gradient signal across the K updates, providing Monte Carlo coverage of the token space within each stage and partially alleviating the single-trajectory limitation. This varies the optimization support but does not sample alternative trajectory-conditioned likelihoods. Full masking avoids future-token exposure while changing the conditioning distribution, so it does not guarantee an unbiased estimate of $\mathcal{J}^*$.

Uniform token reward (resolved). The per-token weight $w_{g,i}$ assigns $\lambda_{\mathrm{s}} > 1$ to structural tokens and $\lambda_{\mathrm{r}} < 1$ to refinement tokens, replacing the uniform $w_{g,i} = 1$ used by both existing families. This amplifies gradient updates on global composition and attenuates updates on near-deterministic local detail, following the rationale in Section 3.2. Trajectory recording methods avoid stage conflation and future-token contamination but leave credit assignment uniform. HT-GRPO is therefore the only paradigm that addresses all four limitations.

B.4 Comparison

Table 4 summarizes the three paradigms along eight dimensions. The formal basis for the entropy-related entries is provided in Section C (Propositions C.1–C.3). Among the four limitations identified in Section 2.3, random remasking methods introduce stage conflation and future-token contamination while achieving only partial path coverage. Trajectory recording methods avoid the first two but sacrifice path diversity and require $O(T)$ forward passes per rollout. Neither family addresses the uniform token reward problem. HT-GRPO uses separate Structure and Refinement updates after the joint Global stage. Full masking eliminates future-token exposure, random subsets vary optimization support, and generation-order-aware weights replace uniform credit assignment. These choices do not recover trajectory-conditioned ratios or establish unbiasedness.

Table 4: Comparison of three dMLLM RL paradigms under a unified formulation. Limitation rows correspond to the four issues identified in Section 2.3.

Dimension	Random Remasking	Trajectory Recording	HT-GRPO
Design choices			
Support set $\mathcal{M}_g^{(k)}$	Remasked positions	All N tokens	Random subset of active stage
Conditioning context $\mathbf{C}_{g,i}^{(k)}$	Randomly retained tokens after remasking	True rollout state $\mathbf{x}_g^{(\prec i)}$	Fully masked state $\mathbf{C}_\emptyset$
Token weight $w_{g,i}$	Uniform ($= 1$)	Uniform ($= 1$)	Generation-order-aware ($\lambda_{\mathrm{s}}{>}1$, $\lambda_{\mathrm{r}}{<}1$)
Operational cost			
Trajectory storage	Final image only	All T states, or reconstruction from step map	Generation-order ranks $\rho_{g,i}$
Forward passes per inner loop	1	T	1
Limitation analysis			
Stage conflation	Yes	Contexts follow rank	Separated after Global stage
Future-token contamination	Yes	No	Eliminated
Multi-path coverage	Partial (causally inconsistent)	Limited (single trajectory)	Token subsets vary; context fixed
Uniform token reward	Unresolved	Unresolved	Resolved

C Information-Theoretic Basis of Hierarchical Token Grouping and the Prompt-Conditioned Estimator

We establish three formal results underpinning HT-GRPO's design.

Assumption (Approximate Consistency). Let V_i be the token at position i, p_θ the joint generation distribution, and $\mathbf{C}^{[n]}$ any partially revealed context with n positions unmasked and position i still

masked. We assume the model's prediction matches the true conditional:

$$\pi_\theta\Big(\cdot \mid \mathbf{C}^{[n]}, c\Big) = p_\theta\Big(\cdot \mid \mathbf{C}^{[n]}, c\Big). \tag{21}$$

This connects the model's output to standard information-theoretic quantities. We write $H(\pi_\theta(\cdot \mid \mathbf{C}, c))$ for the **predictive entropy**, which measures how uncertain the model is about position i given visible context $\mathbf{C}$. Propositions C.1 and C.3 additionally require that $\mathbf{C}^{[n']}$, where $n' > n$, is obtained from $\mathbf{C}^{[n]}$ by revealing more tokens along the same trajectory while keeping position i masked.

Proposition C.1 (Entropy Monotonicity). *Seeing more context reduces uncertainty.* Under Assumption (21), for any position i and any $0 \le n < n' \le N-1$,

$$\mathbb{E}_{\mathbf{C}^{[n]}}\Big[H\Big(\pi_\theta(\cdot \mid \mathbf{C}^{[n]}, c)\Big)\Big] \ge \mathbb{E}_{\mathbf{C}^{[n']}}\Big[H\Big(\pi_\theta(\cdot \mid \mathbf{C}^{[n']}, c)\Big)\Big]. \tag{22}$$

Proof. Let $\mathbf{D}$ be the additional tokens revealed between $\mathbf{C}^{[n]}$ and $\mathbf{C}^{[n']}$. Since mutual information is non-negative, $I(V_i; \mathbf{D} \mid \mathbf{C}^{[n]}, c) = H(V_i \mid \mathbf{C}^{[n]}, c) - H(V_i \mid \mathbf{C}^{[n']}, c) \ge 0$. Applying Assumption (21) to both sides yields the stated inequality. □

The proposition concerns expected conditional entropy for a fixed position under nested contexts and the consistency assumption. It does not prove monotonicity along every realized trajectory, equal entropy across different positions, or that generation rank measures causal contribution. Tokens unmasked in the same step share a context but can have different predictive entropies. Generation rank is therefore a practical grouping proxy, with aggregate empirical evidence in Appendix I.3.

Definition (Future-token set). For rollout g and position i, define $\mathcal{F}_i^{(g)} = \{j : \rho_{g,j} > \rho_{g,i}\}$ as the set of tokens unmasked *after* i. These tokens are still masked when i is predicted; if any appear in the conditioning context, they inject future information into the current optimization step.

Proposition C.2 (Future-Token Contamination). *Random remasking exposes future tokens with nonzero probability, misspecifying the conditioning distribution.* In random remasking methods, each position is independently remasked with probability $p_k \in (0, 1)$. For any non-final position i, at least one future token from $\mathcal{F}_i^{(g)}$ remains visible in $\mathbf{C}_{g,i}^{(k)}$ with probability

$$\Pr\big[\exists j \in \mathcal{F}_i^{(g)} : \ j \text{ is visible in } \mathbf{C}_{g,i}^{(k)}\big] = 1 - p_k^{|\mathcal{F}_i^{(g)}|} > 0. \tag{23}$$

In the true trajectory context $\mathbf{x}_g^{(\prec i)}$ all future positions are masked; when contamination occurs, the ratio $\tilde{r}_{g,i}(\theta) = \pi_\theta(v_{g,i} \mid \mathbf{C}_{g,i}^{(k)}, c)/\pi_{\theta_{\text{old}}}(v_{g,i} \mid \mathbf{C}_{g,i}^{(k)}, c)$ is evaluated under a causally inconsistent context, leading to a misspecified policy gradient estimator.

Proof. The $|\mathcal{F}_i^{(g)}|$ future positions are remasked independently; all are correctly masked with probability $p_k^{|\mathcal{F}_i^{(g)}|}$, so the complement gives the contamination probability. In $\mathbf{x}_g^{(\prec i)}$, all future positions are masked by definition, confirming the context mismatch. □

Thus, random remasking methods evaluate the likelihood ratio under a conditioning distribution that differs from the trajectory-consistent one.

Remark. Let $r_{g,i}^*(\theta) = \pi_\theta(v_{g,i} \mid \mathbf{x}_g^{(\prec i)}, c)/\pi_{\theta_{\text{old}}}(v_{g,i} \mid \mathbf{x}_g^{(\prec i)}, c)$ denote the trajectory-consistent ratio, where $\mathbf{x}_g^{(\prec i)}$ is the true denoising state just before token i is revealed (Section B). The claim above concerns the *conditioning distribution*, not a guaranteed pointwise inequality: $\tilde{r}_{g,i}$ and $r_{g,i}^*$ could coincide numerically for specific θ, but whenever contamination occurs the estimator is structurally inconsistent with the trajectory-consistent objective.

Unlike $\mathbf{x}_g^{(\prec i)}$, $\mathbf{C}_\emptyset$ additionally masks past tokens, so HT-GRPO does not recover $r_{g,i}^*$. This is a deliberate design choice: $r_{g,i}^*$ can be evaluated for a recorded trajectory, whereas averaging over all alternative trajectories is combinatorially intractable (Section 3). HT-GRPO instead defines a *unified* estimator using the same context $\mathbf{C}_\emptyset$ for every token:

$$\hat{r}_{g,i}(\theta) = \frac{\pi_\theta(v_{g,i} \mid \mathbf{C}_\emptyset, c)}{\pi_{\theta_{\text{old}}}(v_{g,i} \mid \mathbf{C}_\emptyset, c)}. \tag{24}$$

Table 5: **Ablation study on random-subset annealing**. All experiments use $\alpha = 0.3$ and $n_g : n_s : n_r = 2 : 4 : 2$.

Coverage strategy	$(\gamma_{\max}, \gamma_{\min}, \phi)$	Single	Two Obj.	Count	Color	Pos.	Attr.	Overall
Full coverage	(1.0, 1.0, `constant`)	100.00	87.88	72.50	**92.55**	64.00	62.00	79.82
Fixed sparse sampling	(0.2, 0.2, `constant`)	97.50	85.86	75.00	87.23	70.00	**67.00**	80.43
Ascending annealing	(1.0, 0.5, `up`)	100.00	91.92	76.25	88.30	72.00	66.00	82.41
Linear decay (ours)	(1.0, 0.5, `down`)	**100.00**	**92.93**	**77.50**	90.43	**73.00**	66.00	**83.31**

This estimator has three properties. First, it is contamination-free by construction. Second, it is self-consistent: the conditioning context is identical for both policies, ensuring a well-defined likelihood ratio. Third, it is entropy-preserving: predictive entropy under $\mathbf{C}_\emptyset$ is not smaller in expectation than under contexts that condition on additional structural token information (Proposition C.3). This conditional-entropy comparison does not establish a lower bound on rollout reward variance or group-relative advantages.

Proposition C.3 (Entropy Lower Bound under Full Masking). *Revealing structural tokens reduces uncertainty over refinement tokens.* Let $\mathbf{S} = \{V_j : j \in \mathcal{S}_{g,\text{structure}}\}$ denote the structural token values and $\mathbf{x}^{\text{structure}}(\mathbf{S})$ the context revealing only those positions. For any refinement token $i \notin \mathcal{S}_{g,\text{structure}}$,

$$H(\pi_\theta(\cdot \mid \mathbf{C}_\emptyset, c)) \geq \mathbb{E}_{\mathbf{S}}\left[H\left(\pi_\theta(\cdot \mid \mathbf{x}^{\text{structure}}(\mathbf{S}), c)\right)\right]. \tag{25}$$

Proof. By Assumption (21), the left side equals $H(V_i \mid c)$ and the right side equals $H(V_i \mid \mathbf{S}, c)$. Non-negativity of $I(V_i; \mathbf{S} \mid c) = H(V_i \mid c) - H(V_i \mid \mathbf{S}, c) \geq 0$ gives the result. □

Revealing structural tokens can reduce expected predictive entropy under the consistency assumption. This concerns likelihood evaluation, not the rewards of already sampled rollouts. The conditioning ablation supports using $\mathbf{C}_\emptyset$ empirically, but does not establish that full masking prevents reward-variance collapse.

D Standard GRPO Objective

Given a prompt q, GRPO [26] samples G complete outputs $\{o_i\}_{i=1}^G$ from the behavior policy $\pi_{\theta_{\text{old}}}$ and maximizes the clipped surrogate objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\,\{o_i\}\sim\pi_{\theta_{\text{old}}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\,\hat{A}_i,\ \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_i\Big) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\Bigg], \tag{26}$$

where the importance ratio

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q,\, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q,\, o_{i,<t})} \tag{27}$$

measures the relative likelihood of token $o_{i,t}$ under the current policy versus the behavior policy, given the same causal prefix $o_{i,<t}$. The group-relative advantage

$$\hat{A}_i = \frac{R_i - \text{mean}\big(\{R_j\}_{j=1}^G\big)}{\text{std}\big(\{R_j\}_{j=1}^G\big) + \delta} \tag{28}$$

is a scalar derived from the outcome reward R_i and broadcast uniformly to all tokens in output o_i. In practice, the same sampled batch $\{o_i\}$ is reused for K successive gradient steps, with all importance ratios computed relative to the fixed behavior policy $\pi_{\theta_{\text{old}}}$.

E Ablation Study Details

All ablations run on MMaDA with $K = 8$ and $\alpha = 0.3$ unless otherwise stated.

A1: Staged RL Training is Necessary for dMLLMs. Figure 4(a) compares various budget allocation strategies under a fixed inner-loop budget $K = 8$. Both single-stage and two-stage variants yield limited overall performance, scoring between 76.6 and 80.2. The full three-stage Sketch-Then-Paint schedule (Global→Structure→Refinement with a 2:4:2 budget) breaks this bottleneck, reaching 83.3 overall, confirming that the coarse-to-fine ordering itself—not merely token-group separation—drives the gain.

A2: Structure-biased Budget Allocation Maximizes Returns. Figure 4(a) also explores budget distribution across stages. The default 2:4:2 allocation achieves the best overall score of 83.3. Refinement-biased (2:2:4) and global-biased (4:2:2) allocations underperform, indicating that structural tokens require more optimization steps to resolve layout and compositional relations.

A3: Superiority of Coarse-to-fine Annealing Schedule. Table 5 evaluates how the annealing schedule controls token sampling within each stage. Static strategies—constant full coverage ($\gamma_{\max}$=$\gamma_{\min}$=1.0, 79.82) and fixed sparse sampling ($\gamma_{\max}$=$\gamma_{\min}$=0.2, 80.43)—both underperform dynamic variants. Ascending annealing (γ: 0.5→1.0, 82.41) improves over static baselines but lags behind our method: starting with sparse coverage causes unstable gradient directions in early updates. Our linear decay schedule (γ: 1.0→0.5) achieves the best overall score of **83.31**.

A4: Sensitivity to Structure Ratio α. Figure 4(b) evaluates the structural-group boundary. Performance peaks at $\alpha = 0.3$ (83.3). A smaller ratio ($\alpha = 0.1$) leaves many layout-relevant tokens under-optimized (80.2), while a larger ratio ($\alpha = 0.5$) dilutes the hierarchy by including tokens with rich visual context, both degrading performance.

A5: Ablation on Credit Weighting & Ratio Conditioning. Figure 4(c) examines the two components of Section 3.2 and Section 3.1.2. Discarding credit weights ($\lambda_s = \lambda_r = 1$) drops the overall score to 80.8, confirming that amplifying structural-token updates is critical for layout composition. Replacing $\mathbf{C}_\emptyset$ with revealed structural contexts yields 80.6. This supports the fully masked context empirically. Proposition C.3 concerns expected predictive entropy and does not by itself establish ratio-variance or reward-variance collapse.

F Implementation Supplement

F.1 Random-Subset Scheduling Functions

Let $k_s \in \{0, \dots, n_s - 1\}$ denote the within-stage update counter for stage s. The general sampling rate is

$$\gamma_{k_s}^{(s)} = \gamma_{\min}^{(s)} + \left(\gamma_{\max}^{(s)} - \gamma_{\min}^{(s)}\right) \cdot \phi\left(\frac{k_s}{\max(1, n_s - 1)}\right), \tag{29}$$

where $\phi : [0, 1] \to [0, 1]$ is a shape function. The main text uses $\phi(p) = 1-p$ (`down` mode), which specializes to Eq. (8):

$$\gamma_{k_s}^{(s)} = \gamma_{\min}^{(s)} + \left(\gamma_{\max}^{(s)} - \gamma_{\min}^{(s)}\right) \cdot \frac{\max(1, n_s - 1) - k_s}{\max(1, n_s - 1)}. \tag{30}$$

We additionally support the following variants (Table 6):

Table 6: **Scheduling functions for random-subset annealing.**

Mode	$\phi(p)$	Behavior
`down`	$1 - p$	Linear decay from high coverage to low coverage
`up`	p	Linear growth from low coverage to high coverage
`constant`	$\frac{1}{2}$	Fixed coverage independent of progress

Each stage maintains its own $\gamma_{\max}^{(s)}$ and $\gamma_{\min}^{(s)}$, allowing different coverage ranges for the global, structural, and refinement stages.

F.2 Hyperparameter Settings

Table 7 lists all hyperparameters used in our experiments. HT-GRPO is implemented on top of the MaskGRPO codebase, inheriting its optimizer, KL penalty coefficient, and RL loop without modification; only the HT-GRPO-specific settings below are new.

Table 7: **Hyperparameter settings used in all experiments.**

Category	Hyperparameter	Value
HT-GRPO	Structure ratio α	0.3
	Stage budget $n_{\text{global}} : n_{\text{structure}} : n_{\text{refinement}}$	$2 : 4 : 2$
	Structural weight λ_{s}	1.5
	Refinement weight λ_{r}	0.5
Annealing	Schedule mode ϕ	`down`
	$\gamma_{\max}$ (all stages)	1.0
	$\gamma_{\min}$ (all stages)	0.5
RL training	Rollouts per prompt G	9
	Classifier-free guidance scale	3.5
	Reward	HPSv3 + CLIP (ViT-L/14) + UniRwd
Hardware	GPUs	8× A100-80G

G Qualitative Results

G.1 DPG-Bench: Counting Accuracy

Figure 6 compares MaskGRPO and HT-GRPO on two DPG-Bench prompts that require reliable counting on Lumina-DiMOO. The first prompt requires a person pointing toward a trio of birds, where the model must preserve both the object count and the human–object spatial relation. The second prompt requires exactly three eggplants arranged on a table. MaskGRPO often produces an incorrect number of target objects, while HT-GRPO more consistently satisfies the requested count.

This illustrates two benefits of the Sketch-Then-Paint hierarchy. First, counting is treated as a structural decision rather than a refinement detail. HT-GRPO updates all tokens in the Global stage and then focuses on structural tokens in the Structure stage, allowing the model to determine how many objects should appear and where they should be placed before local appearance is refined. Second, the staged inner loop prevents counting-related structural decisions from being diluted by refinement updates. MaskGRPO can mix structure and texture tokens in the same update, so the learning signal for object count and spatial placement competes with color and surface-detail optimization. HT-GRPO separates these roles by optimizing structure before refinement, making the correct count and layout more stable across samples.

G.2 DPG-Bench: Scene-Level Structural Completeness

Figure 7 compares MaskGRPO and HT-GRPO on two DPG-Bench prompts that require coherent scene-level structure and foreground–background composition. For the first prompt, both methods can generate the pigeon, oak tree, and village background, but MaskGRPO shows weaker spatial grounding: the tree roots are not always naturally connected to the ground, and the foreground tree can appear flattened against the village backdrop. The pigeon is also sometimes disproportionately large relative to the tree and cottages, further weakening the scene hierarchy. HT-GRPO better preserves the spatial relationship among the pigeon, oak branches, tree trunk, roots, and village background, producing a more coherent foreground–background structure.

For the second prompt, which requires a parrot with fully outstretched wings flying above hills and forest, MaskGRPO can generate the target bird but often renders it with a flatter, less three-dimensional foreground structure. Its wing geometry and flying pose are less naturally integrated with the landscape. HT-GRPO more reliably preserves the outstretched wing structure, flying pose, and background scene context, resulting in a stronger sense of depth and scene-level coherence. These

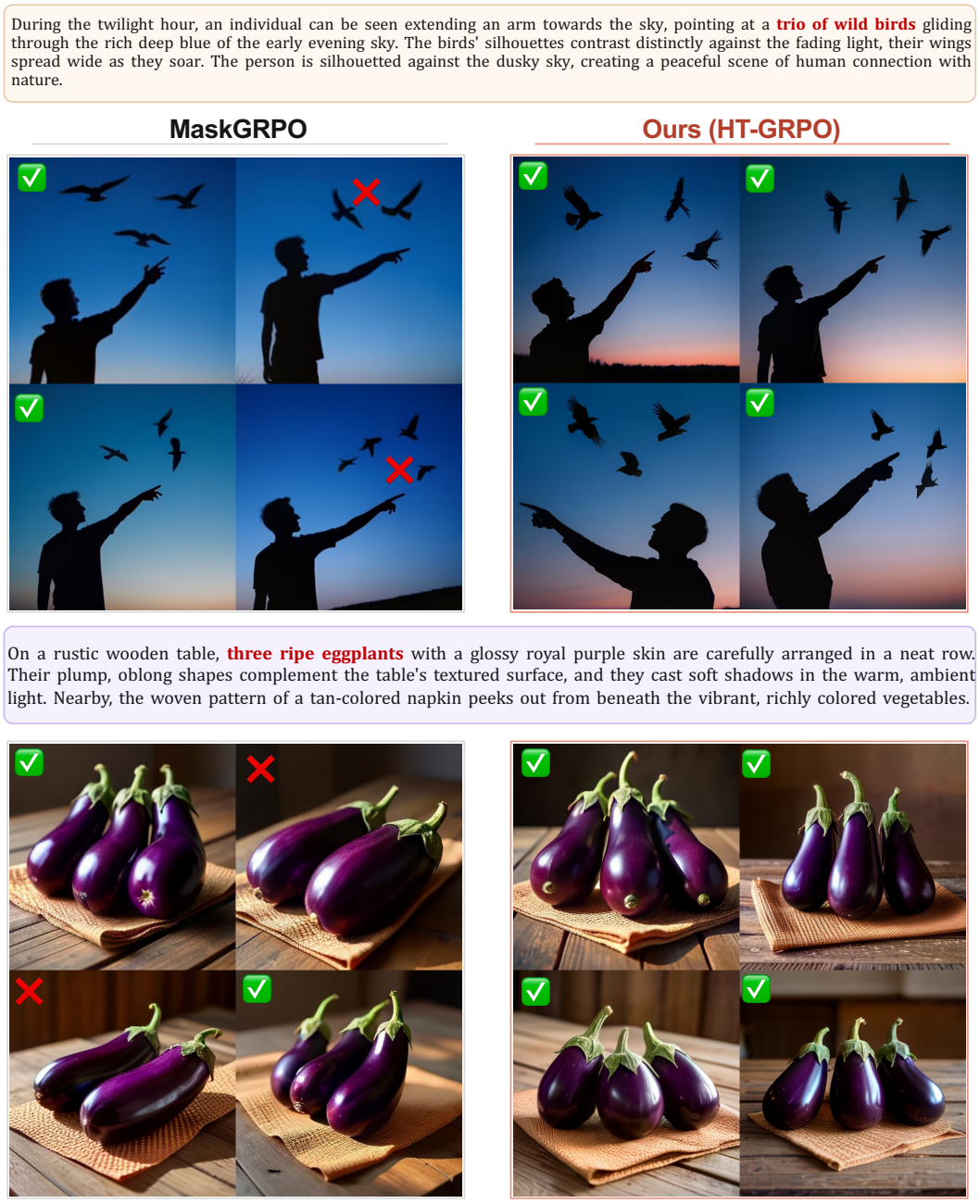


Figure 6: **DPG-Bench counting examples on Lumina-DiMOO.** For each prompt, we compare four samples per method: MaskGRPO on the left and HT-GRPO on the right. HT-GRPO more consistently preserves the requested object count.

examples suggest that the Sketch-Then-Paint hierarchy helps establish global scene layout and object configuration before local appearance refinement.

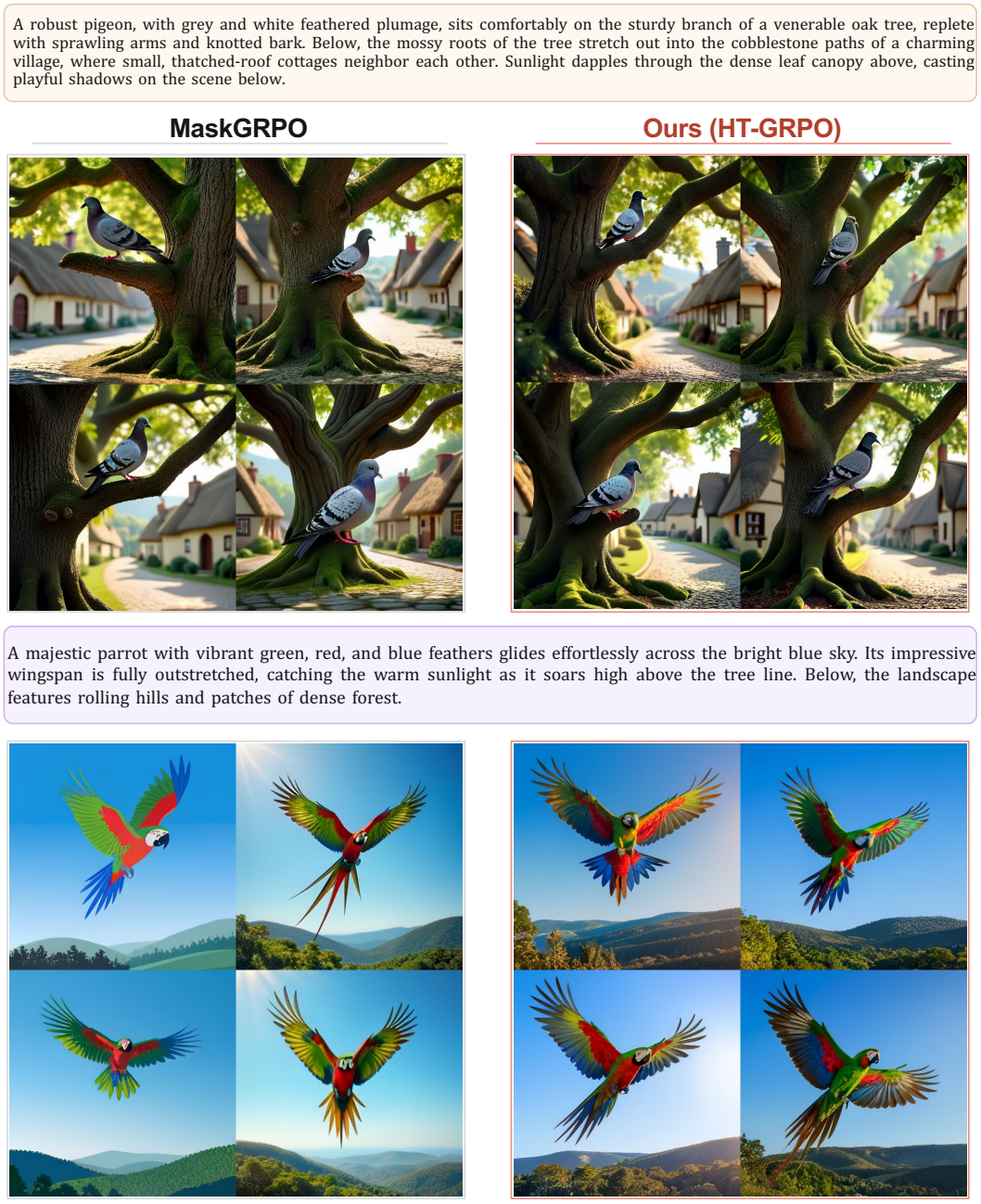


Figure 7: **DPG-Bench scene-level structural completeness on Lumina-DiMOO.** Each prompt compares MaskGRPO and HT-GRPO with four samples per method. HT-GRPO better preserves relative scale, foreground–background structure, and subject pose across complex scene descriptions.

G.3 GenEval: Spatial Grounding and Visual Fidelity

Figure 8 compares the base model, MaskGRPO, and HT-GRPO on representative GenEval examples covering single-object generation, object counting, and spatial relation understanding. Beyond improving correctness, HT-GRPO also produces more natural and visually diverse images. In the

single-object example, all methods generate a baseball bat, while HT-GRPO renders a cleaner object with a more realistic scene composition. In the counting example, HT-GRPO preserves the requested three buses while producing a more diverse urban scene. The spatial-relation example is particularly illustrative: MaskGRPO tends to entangle the book and laptop into a single top-down composition, making the "book above laptop" relation ambiguous. In contrast, HT-GRPO separates the two objects into a clear vertical arrangement, placing the book above the laptop and enriching the scene with natural contextual elements. This suggests that HT-GRPO improves not only object-level accuracy, but also spatial grounding and aesthetic scene construction.

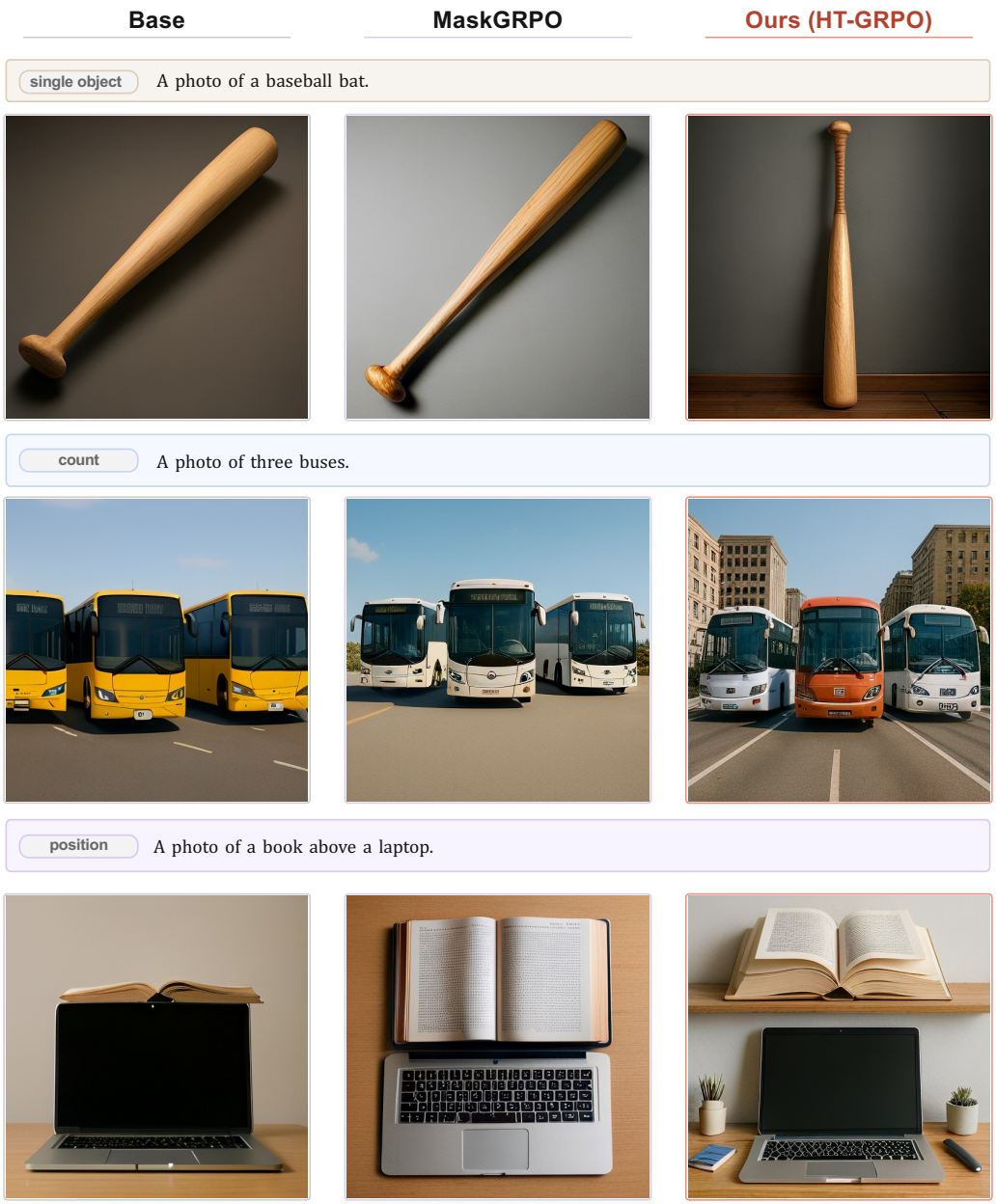


Figure 8: **GenEval qualitative comparison.** From left to right: base model, MaskGRPO, and HT-GRPO. HT-GRPO improves visual fidelity, object counting, and spatial relation grounding while producing more natural scene compositions.

H Complete Training Algorithm

We propose HT-GRPO Training in Algorithm 1. For completeness, the entire training pipeline is summarized in a step-by-step form.

Algorithm 1 HT-GRPO Training

Require: Policy π_θ, reference policy π_{ref}, reward function $R(\cdot)$, prompts $\{c\}$
Require: Structure fraction α, stage budgets $n_{\text{global}}, n_{\text{structure}}, n_{\text{refinement}}$
Require: Annealing rates $\gamma_{\max}^{(s)}, \gamma_{\min}^{(s)}$, credit weights $\lambda_{\text{s}}, \lambda_{\text{r}}$, clip ratio ϵ, KL coefficient β, stability constant δ

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$
  // Phase 1: Rollout
  for $g = 1, \ldots, G$ **do**
    Sample $\mathbf{x}_g^{(0)}$ from $\pi_{\theta_{\text{old}}}$; record unmasking rank $\rho_{g,i}$ for each token i
    Compute reward $R_g \leftarrow R(\mathbf{x}_g^{(0)}, c)$
  end for
  // Phase 2: Token Partitioning
  $N_s \leftarrow \lfloor \alpha N \rfloor$
  $\mathcal{S}_{g,\text{structure}} \leftarrow \{i \mid \rho_{g,i} \le N_s\}, \quad \mathcal{S}_{g,\text{refinement}} \leftarrow \{i \mid \rho_{g,i} > N_s\}, \quad \mathcal{S}_{g,\text{global}} \leftarrow \{1, \ldots, N\}$
  // Phase 3: Reward & Hierarchical Credit Assignment
  $A_g \leftarrow \big(R_g - \text{mean}(\{R_j\})\big) \,/\, \big(\text{std}(\{R_j\}) + \delta\big)$
  $\tilde{A}_{g,i} \leftarrow \begin{cases} A_g \cdot \lambda_{\text{s}} & i \in \mathcal{S}_{g,\text{structure}} \\ A_g \cdot \lambda_{\text{r}} & i \in \mathcal{S}_{g,\text{refinement}} \end{cases}$
  // Phase 4: Sketch-Then-Paint Staged Optimization
  for $s \in$ [Global, Structure, Refinement] **do**
    for $k_s = 0, \ldots, n_s - 1$ **do**
      $\gamma_{k_s}^{(s)} \leftarrow \gamma_{\min}^{(s)} + \big(\gamma_{\max}^{(s)} - \gamma_{\min}^{(s)}\big) \cdot \dfrac{\max(1,\, n_s - 1) - k_s}{\max(1,\, n_s - 1)}$
      for $g = 1, \ldots, G$ **do**
        Sample $\mathcal{M}_g^{(k_s)} \subseteq \mathcal{S}_{g,s}$ with rate $\gamma_{k_s}^{(s)}$
      end for
      Compute $\pi_\theta(v_{g,i} \mid \mathbf{C}_\emptyset, c)$ for all $i \in \bigcup_g \mathcal{M}_g^{(k_s)}$ ▷ single forward pass
      $r_{g,i}(\theta) \leftarrow \pi_\theta(v_{g,i} \mid \mathbf{C}_\emptyset, c) \,/\, \pi_{\theta_{\text{old}}}(v_{g,i} \mid \mathbf{C}_\emptyset, c)$
      $\mathcal{J}^{(k_s)} \leftarrow \dfrac{1}{G}\sum_{g=1}^{G} \dfrac{1}{|\mathcal{M}_g^{(k_s)}|} \sum_{i \in \mathcal{M}_g^{(k_s)}} \min\big(r_{g,i}\,\tilde{A}_{g,i},\ \text{clip}(r_{g,i}, 1-\epsilon, 1+\epsilon)\,\tilde{A}_{g,i}\big) - \beta\, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$
      $\theta \leftarrow \theta + \eta\, \nabla_\theta \mathcal{J}^{(k_s)}$
    end for
  end for
end for

I Additional Analyses

We provide computational, estimator, entropy, and sensitivity analyses, followed by preliminary language-transfer results.

I.1 Matched Computational Profiling

We profile MMaDA training on 2×H200 GPUs with DeepSpeed ZeRO-2, global batch size 72, $K = 8$ inner updates, $T = 12$ generation steps, classifier-free guidance 3.5, and clipping threshold $\epsilon = 0.2$. These profiling runs use different hardware from the main 8×A100-80G experiments. Table 8 reports the measured update cost and peak memory. The shared generation phase was reported as 1484.5 TFLOPs and 8.36 seconds per cycle. Update costs should not be interpreted as end-to-end speedups or total training compute.

Table 8: Matched MMaDA profiling. TFLOPs and time refer to one inner update. Peak memory is the reported peak during the profiling run.

Method	TFLOPs/update	Time/update (s)	Peak memory (GB)
MaskGRPO	173.7	0.17	18.8
TraceGRPO	1490.8	1.54	31.3
HT-GRPO	173.7	0.17	18.9

All three methods share the same T-step rollout. Their post-rollout ratio evaluation differs: a trajectory-conditioned implementation evaluates up to T states per inner update, whereas HT-GRPO evaluates one fully masked state. This gives post-rollout computation $O(KTF(N))$ versus $O(KF(N))$, where $F(N)$ is one model evaluation at sequence length N. Trajectory states can be cached at $O(NT)$ storage or reconstructed from an $O(N)$ step map, with a corresponding recomputation tradeoff. HT-GRPO retains the sample and generation-order ranks at $O(N)$ storage. The measured reduction relative to TraceGRPO is 8.58-fold in update FLOPs, 9.06-fold in update latency, and 12.4 GB in peak memory, within this profiling setup.

I.2 Prompt-Conditioned and Trajectory-Consistent Ratios

We compare the prompt-conditioned ratio $\hat{r}$ with the ratio r^* evaluated under the actual recorded trajectory context. The comparison uses five MMaDA rollout batches with global batch size 72, $T = 12$, and $K = 8$. For this diagnostic, $t = 0$ denotes the initial fully masked state and t then increases with generation, unlike the reverse-time indexing in Eq. (1). We measure the log-ratio difference

$$\delta_{g,i} = \log \hat{r}_{g,i} - \log r^*_{g,i}. \tag{31}$$

Sign agreement is the fraction of tokens for which the two log ratios have the same sign. At $t = 0$, their conditioning contexts coincide.

Table 9: Log-ratio gap across generation phases and inner-update stages. MAE denotes mean absolute error of δ.

Generation phase	Mean δ	MAE	Sign agreement
Initial ($t = 0$)	0.0000	0.0000	1.000
Early ($t = 1$–3)	−0.0001	0.0194	0.505
Middle ($t = 4$–7)	−0.0006	0.0228	0.504
Late ($t = 8$–11)	−0.0004	0.0254	0.499
Inner-update stage	**Mean δ**	**MAE**	**Sign agreement**
Global ($k = 0$–1)	0.0003	0.0099	0.503
Structure ($k = 2$–5)	−0.0008	0.0241	0.507
Refinement ($k = 6$–7)	−0.0005	0.0241	0.504

The small signed means show little average directional shift in these batches, but cancellation can hide tokenwise errors. Sign agreement near 0.5 shows that the surrogate is not a faithful tokenwise

substitute for the trajectory-consistent ratio. Neither a small signed mean nor this comparison establishes unbiasedness relative to the ideal objective $\mathcal{J}^*$ or agreement of policy gradients. The conditioning ablation provides separate end-to-end evidence for this design choice: exposing structural context reduces GenEval from 83.31 to 80.65.

I.3 Aggregate Entropy Measurements

For each backbone, we analyze 756 trajectories collected during the first 500 optimizer steps. At each unmasking step, predictive entropy is averaged over newly unmasked tokens. Let $u = N_{\text{unmasked}}/N$ denote the cumulative unmask ratio. For each trajectory, the reported window change is

$$\Delta H = H(u_{\text{hi}}) - H(u_{\text{lo}}). \tag{32}$$

Table 10 reports the average changes and percentile 95% confidence intervals from 10,000 cluster-bootstrap resamples of trajectories.

Table 10: Predictive-entropy changes across unmask-ratio windows. Brackets give 95% cluster-bootstrap intervals.

Window in u	MMaDA ΔH [CI]	Lumina-DiMOO ΔH [CI]
$[0, 0.2]$	$+0.059$ $[-0.031, 0.151]$	$+0.018$ $[-0.049, 0.084]$
$[0.2, 0.4]$	-0.189 $[-0.210, -0.168]$	-0.179 $[-0.193, -0.167]$
$[0.4, 0.6]$	-0.193 $[-0.203, -0.182]$	-0.197 $[-0.206, -0.188]$
$[0.6, 0.8]$	-0.224 $[-0.232, -0.215]$	-0.218 $[-0.225, -0.211]$
$[0.8, 1.0]$	-0.238 $[-0.247, -0.230]$	-0.241 $[-0.247, -0.234]$

The early-window intervals include zero, whereas all intervals from the transition window onward are negative. The measurements support an aggregate decline beginning within $u \in [0.2, 0.4]$, rather than a universal sharp threshold at $u = 0.3$ or monotonicity of every trajectory. The default $\alpha = 0.3$ lies inside this observed transition range. The controlled MMaDA ablation gives GenEval 80.2, 83.3, and 80.0 for $\alpha = 0.1$, 0.3, and 0.5, respectively, supporting the intermediate boundary among these tested values. The conditional-entropy propositions additionally assume predictive consistency with a joint distribution. The aggregate measurements do not independently verify that assumption.

I.4 Credit-Weight Sensitivity and Selection Rationale

We keep the remaining MMaDA settings fixed while varying the structural and refinement weights (Table 11). Structure-prioritized configurations attain 83.3–83.6 on GenEval, above uniform and reversed credit assignments. The default is not the highest-scoring tested pair. Because both relative credit and average weight scale change across these settings, this is local directional evidence rather than a single-factor causal test or a claim of universal parameter insensitivity.

Table 11: Credit-weight sensitivity on MMaDA. GenEval is reported on a 0–100 scale.

Allocation	λ_{s}	λ_{r}	GenEval
Refinement-prioritized	0.5	1.5	77.9
Uniform	1.0	1.0	80.8
Structure-prioritized	1.2	0.8	83.3
Structure-prioritized (default)	1.5	0.5	83.3
Structure-prioritized	1.6	0.4	83.6

The default boundary, stage budget, and weights encode the intended hierarchy: reserve early tokens for structure, spend half the update budget on them, and give them larger credit weights. The same defaults are used for both image-generation backbones without exhaustive per-backbone or per-benchmark searches. This establishes neither a universal optimum nor equality of hyperparameter search budgets across every baseline. Adaptive boundaries and credit assignment remain future work.

I.5 Preliminary Transfer to a Diffusion Language Model

We additionally apply hierarchy-aware optimization to LLaDA-8B-Instruct [42] on GSM8K [49]. Table 12 reports the preliminary accuracies. These results are preliminary: a complete transfer configuration, independent-seed uncertainty, and broader task coverage are not provided, so they do not establish general transfer beyond the evaluated dMLLM image-generation setting. Continuous text-to-image diffusion transfer has not been experimentally demonstrated here.

Table 12: Preliminary GSM8K results on LLaDA-8B-Instruct. The table does not establish fully matched training or evaluation budgets.

Method	Accuracy (%)
LLaDA-8B-Instruct (zero-shot)	76.7
Diffu-GRPO/D1	79.8
TraceRL	82.1
MaskGRPO	84.2
HT-GRPO	85.3

I.6 Broader Impacts and Release Limitations

Better prompt adherence can make image generation more useful, and lower update costs can reduce resources required for post-training. Improved image generation can also facilitate misleading imagery, impersonation, or harmful content, while retaining biases and privacy risks inherited from base models and data. Possible mitigations include content filters, provenance labeling, and clear model-use documentation, whose effectiveness this study does not evaluate. The reported preference scores are automatic metrics, not a new human-subject evaluation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the method and the two evaluated image-generation backbones. Section 5 and Appendix I.2 delimit estimator and transfer claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses fixed grouping and credit weights, ratio approximation, and preliminary transfer evidence. Appendices I.2 and I.5 give the corresponding limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix C states the consistency and nested-context assumptions and provides proofs. The expected-entropy results do not guarantee monotonicity of each realized trajectory.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the algorithm formula in Appendix H and experiment setups in Section 4 and Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Public implementation material is available at https://github.com/Alpha-VLLM/Lumina-DiMOO/tree/main/ht_grpo. Complete instructions and artifacts sufficient to reproduce all main and diagnostic results have not been established.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have presented the details of experiments in the main paper (Section 4) and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Appendix I.3 reports 95% cluster-bootstrap intervals for entropy changes. The main benchmark scores and preliminary transfer results do not include uncertainty across independent training runs, so statistical significance is not established for all main claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide implementation setup (e.g., GPUs) in Section 4 and include details in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics. The work uses existing models and benchmarks, with limitations and potential misuse discussed in Appendix I.6.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix I.6 discusses image-generation quality and compute benefits alongside misinformation, bias, privacy, and misuse risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: All the data, benchmarks, and models involved are common to the field.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original papers that produced the used assets have been cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our codes and checkpoints are released under the Apache 2.0 license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The reported experiments do not involve crowdsourcing or a new human-subject study. Human-preference scores are computed by automatic metrics.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The reported experiments do not involve crowdsourcing or a new human-subject study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Section 4 identifies the dMLLM backbones and learned reward and evaluation models. Appendix I.5 identifies the diffusion language model used in the preliminary reasoning experiment.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.