GR-GRPO: Graph-Diffused Credit for Autoregressive Image RL Alignment

Zheyu Zhang[1*], Peng-Tao Jiang[1*], Tianyi Zheng[1], Jian Zhang[1], Jinwei Chen[1], and Bo Li[1†]

vivo BlueImage Lab, vivo Mobile Communication Co., Ltd., China
{zhangzy0, libra}@vivo.com

Abstract. Reinforcement-learning alignment for autoregressive image generation is bottlenecked by expensive rollouts: in GRPO-style group optimization, sampling G rollouts per prompt improves exploration and within-group credit estimation but multiplies sampling cost. This challenge is amplified by vector-quantized (VQ) tokenization, where nearby codebook IDs in embedding space can be locally substitutable; however, GRPO-style updates provide explicit positive credit only to sampled IDs, leading to insufficient coverage of success-supported alternatives under small-G training. We propose **GR-GRPO** (Graph-Regularized GRPO), which densifies token-level supervision without additional rollouts by diffusing positive evidence from positive rollouts over a precomputed codebook K-NN graph to construct graph-diffused soft targets, and regularizing the policy toward these targets via an auxiliary cross-entropy term. A confidence-adaptive gate modulates diffusion strength based on the policy's logit margin, and a difficulty-aware budget allocation scheme further improves rollout utilization across prompts. Extensive experiments on compositional image generation (GenEval) and preference-style alignment (DrawBench) validate the effectiveness of GR-GRPO. In particular, GR-GRPO achieves a substantially better compute–performance trade-off, surpassing large-group training (G=64) with $\sim 4.6\times$ less rollout time on Janus-Pro-1B. Our code and model weights are available at `https://github.com/vivoCameraResearch/GR-GRPO`.

Keywords: Image Generation · GRPO · Graph Regularization

1 Introduction

Autoregressive (AR) models have become a competitive paradigm for visual generation, scaling effectively to high-resolution synthesis [4, 26, 39, 45, 47]. Aligning these models with human intent or preference signals is increasingly done through reinforcement learning from sequence-level scalar feedback, ranging from discrete success verifiers to continuous preference/aesthetic scorers [7, 11, 18, 28, 58]. A central practical constraint is rollout cost: generating a single image requires

[*] Equal contribution. [†] Corresponding author.

predicting hundreds to thousands of tokens, so on-policy training is dominated by sampling, and the total budget limits how many rollouts can be afforded.

This budget constraint conflicts with how group-based policy optimization is typically made effective. Methods such as GRPO rely on sampling a group of G rollouts per prompt; larger groups often improve learning by expanding trajectory exploration and stabilizing relative credit within the group. However, increasing G directly multiplies rollout cost. As shown in Fig. 1, GenEval performance improves with G, but runtime grows steeply, making large-group training expensive. The key question is therefore: **how can we obtain the coverage benefits of larger groups without paying for more rollouts?**

The challenge is amplified by discrete visual tokenization. Many AR generators operate over a vector-quantized (VQ) codebook, where different code IDs can correspond to similar decoded content, implying local substitutability at the token level, as illustrated in Fig. 3 This property matters for GRPO's credit assignment: under point-wise updates, successful rollouts expose only a few sampled code IDs, while other locally substitutable alternatives receive no explicit credit. With small G, this limited success-support coverage can bias learning toward a narrow subset of token realizations, yielding slower improvement and a persistent performance gap to large-group training. We quantify this effect using **SuccessSetMass** (Fig. 1), the average probability mass that the policy assigns to token IDs observed in successful rollouts. Higher SuccessSetMass indicates broader coverage of success-supported alternatives, while lower values indicate sparse coverage under limited sampling.

Crucially, the VQ codebook provides a concrete handle to address this gap. Fig. 3 shows that replacing tokens in a successful sequence with their codebook K-NN neighbors preserves success and semantic consistency far better than random replacement, indicating strong local substitutability. This suggests that positive credit can be safely shared within a small neighborhood in the codebook embedding space, motivating credit propagation on a codebook graph.

Based on these observations, we propose **GR-GRPO** (Graph-Regularized GRPO) to improve success-support coverage without increasing rollouts. The core idea is to convert sparse hard-hit evidence from successful or high-scoring trajectories into a denser, set-valued supervision signal over locally substitutable code IDs. Concretely, we aggregate positive token evidence and diffuse it on a precomputed codebook K-NN graph to form graph-diffused soft targets; we then regularize the policy toward these targets via an auxiliary cross-entropy term $\mathcal{L}_{\mathrm{GR}}$. A confidence-adaptive gate controls the diffusion strength. Orthogonally, we use a difficulty-aware budget allocation scheme that allocates more sampling to prompts that require more exploration, based on an offline difficulty estimate.

Experiments on Janus-Pro-1B/7B show that GR-GRPO consistently improves both compositional alignment on GenEval and preference-style alignment on DrawBench. On GenEval, graph regularization at G=8 improves the Janus-Pro-1B score from 0.84 to 0.86. It matches vanilla GRPO at G=32 with a 3.3× reduction in rollout time. With difficulty-aware budget allocation, the full system further reaches 0.87 and surpasses GRPO at G=64 with a 4.6× reduction in roll-

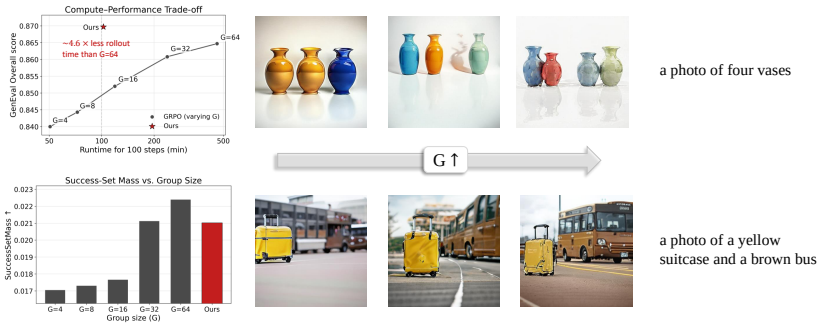

Fig. 1: Compute–coverage trade-off of group sampling on GenEval (Janus-Pro-1B). **Left:** sweeping $G \in \{4, 8, 16, 32, 64\}$ for GRPO, larger G improves GenEval success rate but increases rollout sampling time (top), and increases **SuccessSetMass** (bottom; Sec. 4.3). A higher SuccessSetMass indicates that the policy assigns more probability mass to a broader set of success-supported token alternatives, improving coverage of viable successful options. **Right:** representative generations for the same prompt under GRPO models trained with increasing G. GR-GRPO attains high success rate with high SuccessSetMass at substantially lower rollout cost.

out time (Fig. 1). On DrawBench, GR-GRPO also improves multiple pretrained preference and quality metrics across both model scales.

In summary, our main contributions are:

- We diagnose a compute–coverage bottleneck in group-based RL for VQ-based AR image generation: larger groups improve performance but rapidly increase rollout cost, while small-G training under-covers success-supported token alternatives, as quantified by **SuccessSetMass**.
- We propose **GR-GRPO**, which exploits local token substitutability in the VQ codebook (Fig. 3) and diffuses positive token evidence on a codebook K-NN graph to construct graph-diffused targets, densifying supervision without additional rollouts.
- GR-GRPO improves GenEval and DrawBench across model scales and achieves a substantially better compute–performance trade-off than increasing G alone.

2 Related Work

2.1 Autoregressive Image Generation

Autoregressive (AR) image generation models an image as a token sequence and performs next-token prediction with decoder-only Transformers, inheriting the scaling behavior of large language models. A central design choice is the token space. Many AR generators discretize images into compact visual tokens via vector-quantized (VQ) codebooks, where advances in tokenizers and

discrete representations have substantially improved perceptual fidelity and enabled high-resolution synthesis [9, 14, 29, 30, 33, 37, 40, 42]. Other AR formulations explore masked prediction and training objectives [2, 23]. With stronger text priors and large-scale training, recent AR systems achieve competitive text-conditioned generation [26, 33, 39, 45]. More recently, unified multimodal models increasingly combine understanding and generation within a single decoder-only backbone [4, 21, 47, 59]. Hybrid designs further couple token prediction with diffusion/flow components to leverage complementary inductive biases [6, 50, 65]. In this work, we focus on VQ-based AR generators, where the fixed discrete codebook provides exploitable geometric structure for token-level supervision.

2.2 RL for Image Generation

Reinforcement learning is increasingly used to align text-to-image models beyond maximum-likelihood training. Preference-based objectives such as DPO [32] have been adapted to image generation, including diffusion variants [25, 43, 54] and emerging AR counterparts [13]. More recently, GRPO [36] has also been widely adopted in diffusion/flow [27, 53] and AR image generation [41, 44, 56]. In diffusion/flow settings, GRPO variants often improve efficiency by modifying solver or rollout structure, e.g., branching rollouts, time-windowed updates, to reduce iterative denoising cost [15, 22, 24, 64]. For AR generators, recent work primarily improves learning signal by altering what is optimized within a rollout. Reasoning-augmented approaches introduce explicit CoT-style intermediate structures, such as semantic planning and layout-level reasoning, and optimize both the reasoning process and the final image with RL signals [7, 18]. Token/step-focused approaches improve GRPO-style training by concentrating updates on informative positions, for example by identifying critical tokens or critical refinement steps, and by mitigating training instability such as conflicting token gradients and undesirable entropy dynamics [28, 57, 58]. These approaches still apply ID-specific credit at sampled tokens. In contrast, we diffuse positive credit over a codebook K-NN graph to graph-supported alternatives, improving success-support coverage without additional rollouts.

2.3 Sample Efficiency in Policy Optimization

On-policy optimization for large generative models is often dominated by the cost of collecting rollouts. Existing work improves sample efficiency by selecting, reusing, or prioritizing informative prompts and trajectories [20, 38, 52, 60, 62, 63], expanding exploration through branching or tree-structured rollouts [16, 17], or accelerating generation with speculative decoding [49, 61]. Another direction densifies feedback along sampled trajectories through purpose-designed reward signals, such as token-, step-, or intermediate-level rewards [5, 16].

Our approach is complementary. Rather than modifying reward granularity, changing which trajectories are explored, or accelerating their generation, GR-GRPO retains the original sequence-level rewards and collected rollouts, but densifies token-level supervision by diffusing positive evidence from sampled code

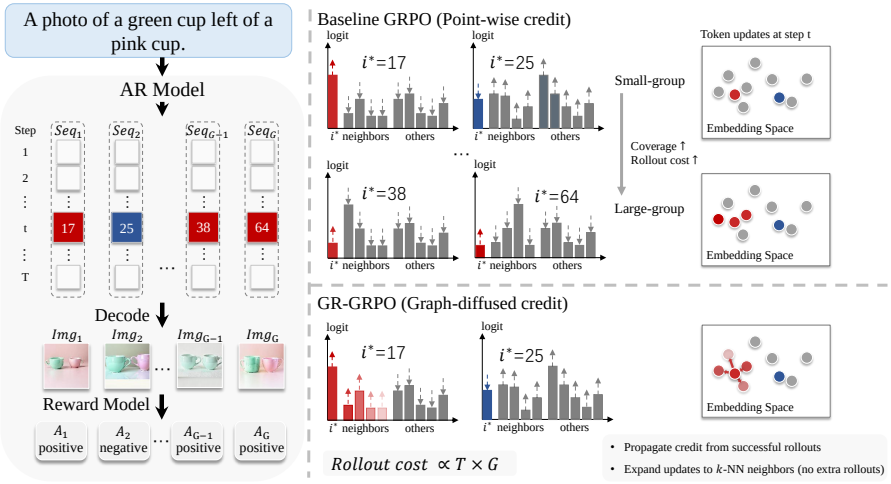

Fig. 2: GR-GRPO overview. Left: for each prompt, we sample G autoregressive token sequences (length T), decode images, and obtain trajectory-level advantages from a reward model. **Right-top (GRPO):** token-level updates provide ID-specific (one-hot) credit on sampled code IDs, leading to sparse supervision under small G. **Right-bottom (GR-GRPO):** we densify positive credit from successful rollouts to K-NN neighbors on a fixed codebook graph to form soft targets, and match them via an auxiliary cross-entropy loss. This expands success-supported alternatives without extra rollouts, keeping rollout cost $\propto T \times G$.

IDs to locally supported, unsampled alternatives. It can therefore be combined with these efficiency techniques.

3 Method

3.1 Preliminary

Autoregressive text-to-image generation. Given a text prompt p, an autoregressive policy π_θ generates a discrete visual-token sequence $x_{1:T} \in \{1, \dots, V\}$ over a fixed codebook of size V:

$$\pi_\theta(x_{1:T} \mid p) = \prod_{t=1}^{T} \pi_\theta(x_t \mid x_{<t}, p). \tag{1}$$

The generated sequence is decoded into an image, which is then scored by a reward model, producing sequence-level feedback $R(x_{1:T}, p) \in \mathbb{R}$ (e.g., a binary success indicator or a continuous preference score).

Group Relative Policy Optimization (GRPO) with KL regularization. For each prompt p, GRPO samples G rollouts $\{x_{1:T}^{(j)}\}_{j=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid p)$ with rewards $\{R^{(j)}\}$

and computes group-relative advantages, e.g.,

$$A^{(j)} = \frac{R^{(j)} - \mathrm{mean}\left(\{R^{(\ell)}\}_{\ell=1}^{G}\right)}{\mathrm{std}\left(\{R^{(\ell)}\}_{\ell=1}^{G}\right)}. \tag{2}$$

The policy is updated by minimizing a clipped surrogate objective:

$$\begin{aligned}\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\ \mathbb{E}_p\, \mathbb{E}_{\{x^{(j)}\}_{j=1}^{G} \sim \pi_{\theta_{\mathrm{old}}}} \Bigg[\frac{1}{G} \sum_{j=1}^{G} \sum_{t=1}^{T} \min\Big(r_t^{(j)}(\theta) A^{(j)}, \\ \mathrm{clip}\big(r_t^{(j)}(\theta), 1-\epsilon, 1+\epsilon\big) A^{(j)} \Big) \Bigg],\end{aligned} \tag{3}$$

where the token-level importance ratio is defined as

$$r_t^{(j)}(\theta) = \frac{\pi_\theta\Big(x_t^{(j)} \mid x_{<t}^{(j)}, p\Big)}{\pi_{\theta_{\mathrm{old}}}\Big(x_t^{(j)} \mid x_{<t}^{(j)}, p\Big)}. \tag{4}$$

To prevent excessive policy deviation and stabilize training, GRPO incorporates a KL-divergence penalty against a reference policy π_{ref}:

$$\mathcal{L}_{\mathrm{KL}}(\theta) = \mathbb{E}_p\, \mathbb{E}_{\{x^{(j)}\}_{j=1}^{G} \sim \pi_{\theta_{\mathrm{old}}}} \left[\frac{1}{G} \sum_{j=1}^{G} \sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x_{<t}^{(j)}, p) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x_{<t}^{(j)}, p)\big) \right]. \tag{5}$$

Token-level credit assignment is ID-specific. A key property of GRPO is the shape of token-level credit induced by the softmax parameterization.

Ignoring clipping and the KL term for intuition, we analyze the unclipped GRPO gradient through an equivalent per-step surrogate loss

$$\ell_t^{(j)}(\theta) \ = \ -A^{(j)} \log \pi_\theta(x_t^{(j)} \mid x_{<t}^{(j)}, p), \tag{6}$$

which matches the token-level policy-gradient structure of Eq. (3). Let $\pi_\theta(\cdot \mid x_{<t}^{(j)}, p) = \mathrm{softmax}(z_t)$ with logits $z_t(i)$. Then

$$\frac{\partial \ell_t^{(j)}(\theta)}{\partial z_t(i)} = A^{(j)} \Big(\pi_\theta(i \mid x_{<t}^{(j)}, p) - \mathbb{I}[i = x_t^{(j)}] \Big), \tag{7}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Therefore, the per-step logit update direction induced by rollout j is proportional to

$$\Delta z_t(i) \ \propto \ A^{(j)} \Big(\mathbb{I}[i = x_t^{(j)}] - \pi_\theta(i \mid x_{<t}^{(j)}, p) \Big), \qquad \forall i \in \{1, \ldots, V\}. \tag{8}$$

Consequently, when $A^{(j)} > 0$ the sampled ID $x_t^{(j)}$ is pushed up, while all non-sampled IDs are pushed down in proportion to their current probabilities; when $A^{(j)} < 0$ the direction reverses. In either case, semantically adjacent IDs receive no direct positive credit unless they are also sampled, which limits success-support coverage under small-group training and motivates our graph-regularized credit densification.

3.2 Bottleneck: Insufficient Success-Support Coverage in Small-Group Regimes

The limitation above becomes particularly acute in VQ-based autoregressive generation. Each token selects a codebook entry with an embedding; nearby embeddings often decode to visually similar content. Thus, token realizations are non-unique: multiple embedding-space K-NN code IDs can be locally substitutable with minimal perceptual change, as shown in Fig. 3.

However, GRPO assigns direct positive credit only to the sampled IDs in successful trajectories. Under tight compute, small-G training yields only a few observed success IDs per prompt and position, under-covering success-supported alternatives. We quantify this gap via **SuccessSetMass** (Fig. 1, Sec. 4.3). While increasing G improves coverage, it linearly amplifies rollout cost, motivating a mechanism that expands success-support coverage without additional sampling.

3.3 Structure-Aware Positive Evidence: From Observed IDs to Graph-Diffused Targets

Key idea. We treat each positive rollout as evidence for a set of locally substitutable code IDs, and propagate this evidence over a fixed codebook graph to obtain sparse soft targets—recovering large-G coverage under small-G rollouts.

To address insufficient success-support coverage (Sec. 3.2), we convert sparse evidence from positive trajectories into graph-diffused soft targets. As illustrated in Fig. 2, baseline GRPO updates only sampled IDs, whereas GR-GRPO diffuses positive evidence to local alternatives on a fixed codebook K-NN graph without additional rollouts. Concretely, we construct the targets through a sparse pipeline: *count* $\rightarrow$ *graph diffusion* $\rightarrow$ *support-restricted normalization*.

(A) Positive evidence aggregation. For a prompt-specific group, let $\mathcal{P} \subseteq \{1, \ldots, G\}$ be the indices of positive rollouts. For discrete verifiers, $\mathcal{P} = \{j \mid R^{(j)} = 1\}$; for continuous rewards, $\mathcal{P}$ contains the top-50% rollouts within the group by reward. At each position t, we form sparse counts $c_t \in \mathbb{R}^V_{\geq 0}$:

$$c_t(i) = \sum_{j \in \mathcal{P}} \mathbb{I}\left[x_t^{(j)} = i\right], \qquad i \in \{1, \ldots, V\}. \tag{9}$$

We restrict diffusion to positive evidence; propagating negative evidence empirically degrades performance (Sec. 4.4), likely because failure modes are heterogeneous and do not form a coherent local neighborhood in codebook space.

(B) Graph diffusion on the codebook K-NN graph. We approximate local token substitutability using a fixed K-NN graph over the codebook embeddings by cosine similarity. Let $e_i \in \mathbb{R}^D$ be the embedding of ID i and $\bar{e}_i = e_i / \|e_i\|_2$. For each ID i, let $\mathcal{N}(i)$ be its top-K nearest neighbors (excluding i). We define diffusion weights

$$a_{ij} = \frac{\exp(\cos(\bar{e}_i, \bar{e}_j))}{\sum_{\ell \in \mathcal{N}(i)} \exp(\cos(\bar{e}_i, \bar{e}_\ell))}, \qquad j \in \mathcal{N}(i), \tag{10}$$

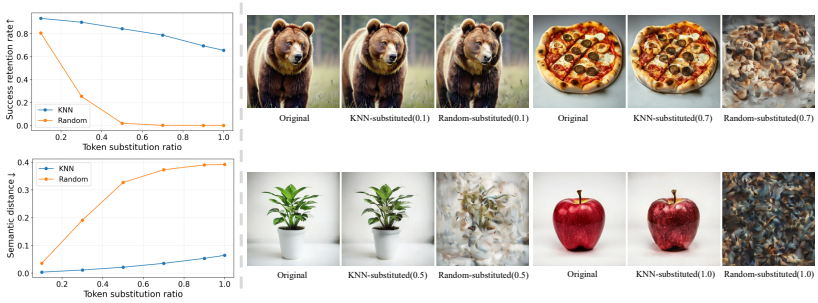

Fig. 3: Local substitutability of VQ tokens. Starting from successful GenEval rollouts, we replace a fraction ρ of tokens with either codebook K-NN neighbors (cosine similarity in codebook-embedding space) or random IDs. We report verifier success retention (top) and CLIP feature distance to the original image (bottom), where `KNN-substituted(`ρ`)` denotes K-NN substitution at token replacement ratio ρ.

so that $\sum_{j\in\mathcal{N}(i)} a_{ij} = 1$. We diffuse observed ids to neighbors:

$$\tilde{c}_t(j) = \sum_{i:\, c_t(i)>0} c_t(i)\, a_{ij} \cdot \mathbb{I}[j \in \mathcal{N}(i)]\,, \qquad j \in \{1,\dots,V\}. \tag{11}$$

This diffusion can be implemented efficiently via neighbor lookup and accumulation.

(C) Support-restricted normalization (graph-diffused target). Define the support

$$U_t \;=\; \{i:\, c_t(i) > 0\} \;\cup\; \mathcal{N}\big(\{i:\, c_t(i) > 0\}\big), \tag{12}$$

i.e., the observed IDs and their K-NN neighbors. An unnormalized mass on U_t:

$$\hat{q}_t(i) = \begin{cases} c_t(i) + \alpha_s\, \tilde{c}_t(i), & i \in U_t, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

We then normalize over U_t to obtain the soft target distribution:

$$q_t(i) = \frac{\hat{q}_t(i)}{\sum_{j\in U_t} \hat{q}_t(j)}, \qquad i \in U_t. \tag{14}$$

We apply $q_t(i)$ only on prefixes of positive rollouts and restrict supervision to the sparse support U_t with confidence-adaptive gating, mitigating over-regularization across heterogeneous prefixes.

3.4 Auxiliary Objective and Confidence-Adaptive Gating

We use the above graph-diffused targets $\{q_t\}$ as additional token-level supervision. Specifically, we introduce a cross-entropy graph-regularization loss that aligns the policy distribution with q_t on the sparse support U_t.

Graph-regularization loss. We apply targets at prefixes of positive rollouts:

$$\mathcal{L}_{\mathrm{GR}}(\theta) = \frac{1}{G}\sum_{j\in\mathcal{P}}\sum_{t=1}^{T} g_t^{(j)} \cdot \mathrm{CE}\Big(q_t, \pi_\theta(\cdot \mid x_{<t}^{(j)}, p)\Big)$$
$$= -\frac{1}{G}\sum_{j\in\mathcal{P}}\sum_{t=1}^{T} g_t^{(j)} \sum_{i\in U_t} q_t(i) \log \pi_\theta\Big(i \mid x_{<t}^{(j)}, p\Big), \tag{15}$$

where $g_t^{(j)} \in [0,1]$. If no positive rollouts are observed, $\mathcal{P} = \emptyset$ and $\mathcal{L}_{\mathrm{GR}} = 0$.

Gradient view (contrast to point-wise credit). Eq. (7) shows that the GRPO surrogate induces a point-wise credit shape with an implicit one-hot target $\mathbb{I}[i = x_t^{(j)}]$. In contrast, Eq. (15) replaces this one-hot target by the graph-diffused distribution q_t supported on U_t. Treating $g_t^{(j)}$ as a coefficient,[1] the per-step cross-entropy term yields the standard target-matching gradient w.r.t. logits:

$$\frac{\partial \mathcal{L}_{\mathrm{GR}}}{\partial z_t(i)} \;\propto\; \frac{1}{G}\sum_{j\in\mathcal{P}} g_t^{(j)}\Big(\pi_\theta(i \mid x_{<t}^{(j)}, p) - q_t(i)\Big), \qquad \forall i \in \{1,\dots,V\}, \tag{16}$$

where $q_t(i) = 0$ for $i \notin U_t$ by construction. Accordingly, under gradient descent on $\mathcal{L}_{\mathrm{GR}}$, the logit update direction is

$$\Delta z_t(i) \;\propto\; \frac{1}{G}\sum_{j\in\mathcal{P}} g_t^{(j)}\Big(q_t(i) - \pi_\theta(i \mid x_{<t}^{(j)}, p)\Big), \tag{17}$$

which mirrors Eq. (8) with $\mathbb{I}[i = x_t^{(j)}]$ replaced by q_t. Thus, $\mathcal{L}_{\mathrm{GR}}$ pulls probability mass toward a set-valued target over success-supported alternatives rather than a single observed id, densifying token-level supervision.

Confidence-adaptive gating. We apply the auxiliary supervision with a confidence-adaptive gate $g_t^{(j)} \in (0,1)$. Let $z_t^{(j)}$ be the logits for prefix $(x_{<t}^{(j)}, p)$ and define the top-1/top-2 margin

$$m_t^{(j)} = z_{t,(1)}^{(j)} - z_{t,(2)}^{(j)}, \tag{18}$$

where $z_{t,(1)}^{(j)} = \max_i z_t^{(j)}(i)$ and $z_{t,(2)}^{(j)}$ is the second-largest logit. A small margin means the top candidates have comparable likelihood under softmax, so the update induced by a single sampled ID is more brittle; in VQ codebooks, these competing candidates are often locally substitutable, making neighborhood-level supervision preferable. Conversely, a large margin indicates a stable dominant choice, where strong diffusion would unnecessarily spread mass to neighbors and over-smooth the update. We gate diffusion strength by

$$g_t^{(j)} = \sigma\left(\frac{\beta - m_t^{(j)}}{\tau}\right), \tag{19}$$

[1] We stop gradients through $g_t^{(j)}$ so that it acts as confidence-dependent reweighting.

so that low-margin steps receive stronger graph-diffused supervision, whereas high-margin steps suppress diffusion to mitigate over-smoothing. We use fixed defaults $(\beta, \tau) = (1, 0.5)$ and report sensitivity in the appendix.

3.5 Overall Training Objective

Our final objective is defined as:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}(\theta) + \lambda_{\text{GR}}\mathcal{L}_{\text{GR}}(\theta). \tag{20}$$

$\mathcal{L}_{\text{GR}}$ complements $\mathcal{L}_{\text{GRPO}}$ by encouraging probability mass on graph-diffused success-supported alternatives, mitigating ID-specific credit sparsity under small-group sampling. It incurs negligible overhead since it is computed on sparse supports with precomputed codebook neighbors.

3.6 Difficulty-Aware Budget Allocation

We allocate rollouts with a difficulty-aware prompt-dependent schedule $G(p)$. Prompts are scored by the initial policy, bucketed by difficulty, and mapped to bin-specific group sizes. This module is orthogonal to graph-diffused credit (it only reallocates rollouts) yet complementary in practice; we enable it by default and ablate its effect in Sec. 4.4, with details in the appendix.

4 Experiments

We evaluate GR-GRPO on two text-to-image alignment settings using the VQ-based autoregressive backbone Janus-Pro [4]. **(i) Compositional alignment (GenEval)** [12]: an official verifier provides a discrete success signal; we report the official success rate (in $[0, 1]$) and use the verifier output as the training reward. **(ii) Preference/quality alignment (DrawBench)** [34]: we train with HPSv2.1 as a continuous reward [48] and evaluate on multiple pretrained preference and quality metrics including HPSv2.1 [48], DeQA [55], PickScore [19], ImageReward [51], Aesthetic score [35], and UnifiedReward [46].

4.1 Experimental Setup

We fine-tune Janus-Pro-1B/7B for 1600 steps on 8× NVIDIA H20 GPUs, using a fixed reference model (initial Janus-Pro weights) for KL regularization with $\lambda_{\text{KL}} = 0.01$. Following prior work [27, 58], we train GenEval alignment on the same 50K-prompt set and train DrawBench alignment on 15K prompts sampled from the HPSv2 training dataset. GR-GRPO uses $\lambda_{\text{GR}} = 0.05$, $K = 64$, and $\alpha_s = 0.5$ with the confidence-adaptive gate in Sec. 3.4, learning rates 3×10^{-6} (GenEval) and 1×10^{-6} (DrawBench), and a difficulty-aware prompt-dependent group-size schedule (details in the appendix).

Table 1: GenEval results. We place comparisons on the Janus-Pro backbone at the top for clarity. Best and second best are highlighted in blue and green, respectively.

Method	Overall↑	Sing Obj.↑	Two Obj.↑	Counting↑	Color↑	Position↑	Color Attr.↑
	Janus-Pro backbone						
Janus-Pro-1B [4]	0.73	0.98	0.82	0.51	0.89	0.65	0.56
Janus-Pro-1B+GRPO	0.84	1.00	0.95	0.59	0.84	0.88	0.77
GCPO-1B [58]	0.85	1.00	0.96	0.63	0.88	0.91	0.73
GR-GRPO-1B (Ours)	0.87	1.00	0.98	0.67	0.89	0.94	0.76
Janus-Pro-7B [4]	0.80	0.99	0.89	0.59	0.90	0.79	0.66
Janus-Pro-7B+GRPO	0.87	0.99	0.92	0.71	0.94	0.92	0.73
GCPO-7B [58]	0.90	0.99	0.95	0.90	0.90	0.95	0.76
STAGE [28]	0.89	0.99	0.95	0.82	0.90	0.89	0.79
GR-GRPO-7B (Ours)	0.92	1.00	0.97	0.91	0.92	0.96	0.74
	Diffusion-based Methods						
PixArt-α [3]	0.48	0.98	0.50	0.44	0.80	0.08	0.07
SDXL [31]	0.55	0.98	0.74	0.39	0.85	0.15	0.23
SD3 [8]	0.63	0.98	0.78	0.50	0.81	0.24	0.52
DALL-E 3 [1]	0.67	0.96	0.87	0.47	0.83	0.43	0.45
	AR-based Methods						
LlamaGen [37]	0.32	0.71	0.34	0.21	0.58	0.07	0.04
Chameleon [39]	0.39	-	-	-	-	-	-
Emu3 [45]	0.54	0.98	0.71	0.34	0.81	0.17	0.21
Show-o [50]	0.68	0.98	0.80	0.66	0.84	0.31	0.50
Transfusion [65]	0.63	-	-	-	-	-	-
Fluid [10]	0.69	0.96	0.83	0.63	0.80	0.39	0.51
Infinity [14]	0.73	-	0.85	-	-	0.49	0.57
	AR-based Methods + RL						
Show-o+PARM [13]	0.69	0.97	0.75	0.60	0.83	0.54	0.53
T2I-R1 [18]	0.79	0.99	0.91	0.53	0.91	0.76	0.65
GoT-R1-7B [7]	0.75	0.99	0.94	0.50	0.90	0.46	0.68
AR-GRPO [56]	0.32	0.79	0.26	0.23	0.59	0.04	0.03
MaskFocus [57]	0.76	1.00	0.91	0.85	0.87	0.42	0.54

4.2 Main Results

Tab. 1 reports same-backbone comparisons on Janus-Pro.[2] GR-GRPO consistently improves Janus-Pro across model scales and outperforms strong RL baselines. On Janus-Pro-1B, GR-GRPO raises the overall GenEval score from 0.73 to 0.87 (+0.14), surpassing Janus-Pro-1B+GRPO (0.84) and GCPO-1B (0.85), indicating improved compositional binding under limited sampling. On Janus-Pro-7B, GR-GRPO achieves 0.92, improving over the pretrained Janus-Pro-7B (0.80) and surpassing Janus-Pro-7B+GRPO (0.87), STAGE (0.89).

Tab. 2 evaluates preference/quality alignment using multiple pretrained preference and quality metrics. GR-GRPO yields consistent improvements over the Janus-Pro backbones and RL baselines under continuous-reward training.

Fig. 4 provides representative generations on GenEval and DrawBench for the Janus-Pro-7B backbone. GR-GRPO produces more faithful attribute binding and maintains stronger visual realism across diverse prompts.

[2] We additionally evaluate on other AR backbones in the appendix

Table 2: DrawBench results.

Method	HPSv2.1↑	DeQA↑	PickScore↑	ImageReward↑	Aesthetic↑	UniReward↑
Janus-Pro-1B [4]	25.63	3.58	21.55	0.54	5.89	2.76
Janus-Pro-1B+GRPO	27.22	3.72	21.71	0.73	6.04	2.80
GCPO-1B [58]	27.18	3.71	21.72	0.71	6.05	2.82
GR-GRPO-1B (Ours)	27.55	3.76	21.73	0.74	6.06	2.85
Janus-Pro-7B [4]	26.74	3.61	22.06	0.87	5.90	3.02
Janus-Pro-7B+GRPO	27.52	3.68	22.11	0.92	5.99	3.06
GCPO-7B [58]	26.84	3.59	22.04	0.84	5.87	3.01
STAGE [28]	27.76	3.70	22.19	0.96	6.00	3.10
GR-GRPO-7B (Ours)	28.25	3.74	22.21	0.99	6.06	3.07

4.3 Analysis: Why Small-Group GRPO Underutilizes VQ Redundancy

We present an evidence chain motivating GR-GRPO. (i) Increasing group size G improves GenEval but sharply increases rollout runtime (Fig. 1), making large-G training costly. (ii) The small-G performance gap is accompanied by insufficient *success-support coverage*: **SuccessSetMass** increases with G (Fig. 1), indicating that larger groups expose a broader set of token realizations appearing in positive rollouts. (iii) Crucially, this coverage deficit can be addressed structurally under VQ tokenization: many viable alternatives concentrate in local codebook neighborhoods, and K-NN substitutions preserve verifier success and CLIP-based semantic fidelity far better than random replacement (Fig. 3). These observations motivate propagating positive evidence on a fixed codebook graph to improve coverage without additional rollouts.

(1) Compute–performance trade-off. We sweep $G \in \{4, 8, 16, 32, 64\}$ and report GenEval success rate and rollout runtime. Larger groups improve performance but increase sampling cost steeply (Fig. 1), motivating higher utility per rollout in small-G regimes.

*(2) Why larger G helps: success-support coverage (**SuccessSetMass**).* We hypothesize that larger G improves performance by expanding success-support coverage, i.e., exposing a broader set of token realizations that appear in positive rollouts. Using the same definition as in Sec. 3.3, let $\mathcal{P}$ denote positive rollouts (successful for discrete verifiers; top-50% by reward for continuous rewards) and $S_t = \{x_t^{(j)} : j \in \mathcal{P}\}$. We measure $M_t = \sum_{v \in S_t} \pi_\theta(v \mid x_{<t}, p)$ and report SuccessSetMass $= \frac{1}{T}\sum_{t=1}^{T} M_t$. As shown in Fig. 1, SuccessSetMass increases with G, explaining the small-G gap and indicating that large-G training covers more success-supported alternatives.

(3) Structural source of coverage: local substitutability in VQ tokenization. We validate the structural premise behind coverage expansion in VQ-based AR models. Replacing a fraction of tokens with codebook K-NN neighbors preserves verifier success and CLIP-based semantic fidelity far better than random replacement (Fig. 3), indicating that many success-supported alternatives concentrate

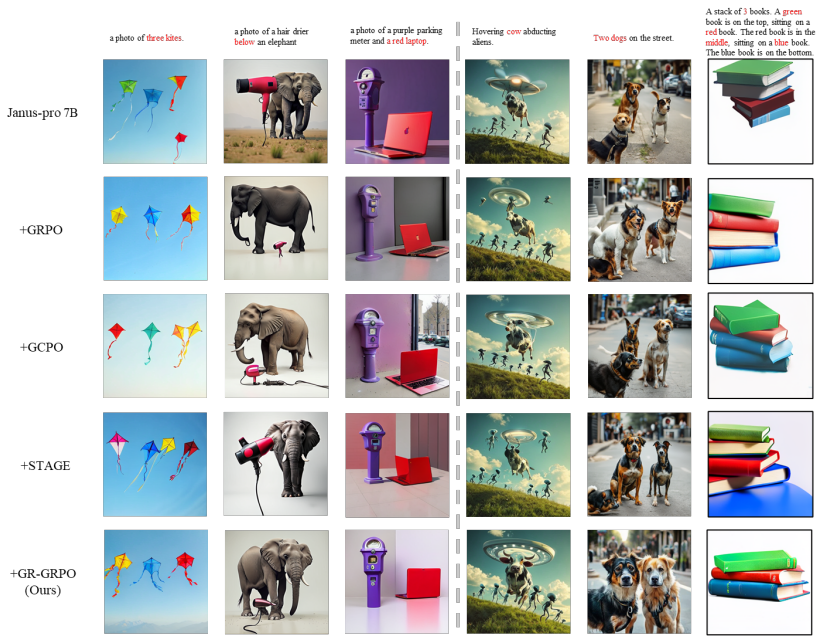

Fig. 4: Qualitative comparisons. The left three columns are GenEval prompts (*Counting*, *Position*, *Color Attribute*), and the right three columns are DrawBench prompts. GR-GRPO better satisfies compositional constraints while preserving visual fidelity.

in local codebook neighborhoods. This supports propagating positive evidence on the codebook graph as a principled mechanism to increase success-support coverage without additional rollouts.

4.4 Ablation Study

All ablations are conducted on GenEval using Janus-Pro-1B.

Component ablation. Tab. 3 isolates the effects of graph regularization and difficulty-aware budget allocation. Budget control alone has a marginal impact on GRPO. In contrast, graph regularization yields a larger improvement ($0.84 \rightarrow 0.86$), with consistent gains on more compositional subsets such as *Two Obj.*, *Counting*, and *Position*. Notably, combining budget control with GR-GRPO achieves the best overall score (0.87), suggesting that better rollout allocation is most effective when paired with graph-diffused credit densification.

Hyperparameter sensitivity. Tab. 4 studies sensitivity to the neighborhood size k, diffusion strength α_s, diffusion scope, and gating. We observe a trade-off in k: too small neighborhoods under-cover valid alternatives, while overly large neighborhoods introduce noise; $K = 64$ performs best among $\{32, 64, 128\}$. Similarly, an intermediate value for α_s is necessary, as overly aggressive diffusion

Table 3: Component ablation on GenEval using Janus-Pro-1B.

Method	Overall↑	Sing Obj.↑	Two Obj.↑	Counting↑	Color↑	Position↑	Color Attr.↑
GRPO	0.84	1.00	0.95	0.59	0.84	0.88	0.77
GRPO + Budget Control	0.84	1.00	0.96	0.60	0.84	0.88	0.76
GR-GRPO w/o Budget Control	0.86	0.99	0.96	0.64	0.88	0.90	0.77
GR-GRPO + Budget Control (Ours)	0.87	1.00	0.98	0.67	0.89	0.94	0.76

Table 4: Sensitivity analyses on GenEval using Janus-Pro-1B.

Factor	Setting	Overall↑	Sing Obj.↑	Two Obj.↑	Counting↑	Color↑	Position↑	Color Attr.↑
K (K-NN)	32	0.84	0.99	0.95	0.61	0.87	0.90	0.75
	64 (Ours)	0.87	1.00	0.98	0.67	0.89	0.94	0.76
	128	0.86	0.99	0.96	0.64	0.89	0.91	0.76
Gating	$g_t \equiv 1$	0.83	0.99	0.93	0.62	0.86	0.86	0.75
	adaptive (Ours)	0.87	1.00	0.98	0.67	0.89	0.94	0.76
Diffusion scope	**pos-only (Ours)**	0.87	1.00	0.98	0.67	0.89	0.94	0.76
	pos+neg (Ours+negative)	0.84	0.99	0.95	0.61	0.88	0.92	0.76
α_s (prop.)	0.3	0.84	0.98	0.95	0.61	0.89	0.90	0.73
	0.5 (Ours)	0.87	1.00	0.98	0.67	0.89	0.94	0.76
	0.7	0.84	1.00	0.96	0.57	0.90	0.89	0.72

tends to over-smooth the supervision signal and degrade compositional accuracy. Furthermore, propagating negative evidence (pos+neg) degrades performance (0.87 → 0.84). Finally, the confidence-adaptive gate is critical: replacing it with a constant gate $g_t \equiv 1$ substantially reduces performance (0.83 vs. 0.87), supporting the need to modulate diffusion strength by policy confidence. Other hyperparameter sensitivity are provided in the appendix.

5 Conclusion

We study RL alignment for VQ-based autoregressive image generation under expensive rollouts. We find that in small-group regimes, GRPO-style updates assign ID-specific token credit and under-cover the many success-supported yet unsampled alternatives induced by VQ redundancy, yielding a compute–coverage bottleneck that manifests as a compute–performance trade-off. To address this without additional rollouts, we propose **GR-GRPO**, which turns sparse positive evidence from positive rollouts into set-valued token supervision by diffusing it on a fixed codebook-embedding K-NN graph to form graph-diffused soft targets and regularizing the policy toward them; a confidence-adaptive gate suppresses diffusion when the policy is already confident. Across Janus-Pro backbones, GR-GRPO improves both GenEval (discrete verifier) and DrawBench (continuous reward) and achieves a substantially better compute–performance trade-off than increasing G alone, e.g., surpassing G=64 with $\sim 4.6\times$ less rollout time on Janus-Pro-1B. Future work includes extending graph-based credit densification beyond VQ codebooks by learning or defining neighborhood structure for continuous-token AR generators and hybrid token–diffusion models.

Acknowledgements

We acknowledge the support from Shanghai Municipal Commission of Economy and Informatization, under Grant No. 2024-GZL-RGZN-01008.

References

1. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf **2**(3), 8 (2023)
2. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11315–11325 (2022)
3. Chen, J., Wu, Y., Luo, S., Xie, E., Paul, S., Luo, P., Zhao, H., Li, Z.: Pixart-{\delta}: Fast and controllable image generation with latent consistency models. arXiv preprint arXiv:2401.05252 (2024)
4. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-pro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811 (2025)
5. Cui, G., Yuan, L., Wang, Z., Wang, H., Zhang, Y., Chen, J., Li, W., He, B., Fan, Y., Yu, T., et al.: Process reinforcement through implicit rewards. arXiv preprint arXiv:2502.01456 (2025)
6. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., et al.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
7. Duan, C., Fang, R., Wang, Y., Wang, K., Huang, L., Zeng, X., Li, H., Liu, X.: Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. arXiv preprint arXiv:2505.17022 (2025)
8. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
9. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
10. Fan, L., Li, T., Qin, S., Li, Y., Sun, C., Rubinstein, M., Sun, D., He, K., Tian, Y.: Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. arXiv preprint arXiv:2410.13863 (2024)
11. Gallici, M., Borde, H.S.d.O.: Fine-tuning next-scale visual autoregressive models with group relative policy optimization. arXiv preprint arXiv:2505.23331 (2025)
12. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. Advances in Neural Information Processing Systems **36**, 52132–52152 (2023)
13. Guo, Z., Zhang, R., Tong, C., Zhao, Z., Huang, R., Zhang, H., Zhang, M., Liu, J., Zhang, S., Gao, P., et al.: Can we generate images with cot? let's verify and reinforce image generation step by step. arXiv preprint arXiv:2501.13926 (2025)
14. Han, J., Liu, J., Jiang, Y., Yan, B., Zhang, Y., Yuan, Z., Peng, B., Liu, X.: Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15733–15744 (2025)

15. He, X., Fu, S., Zhao, Y., Li, W., Yang, J., Yin, D., Rao, F., Zhang, B.: Tempflow-grpo: When timing matters for grpo in flow models. arXiv preprint arXiv:2508.04324 (2025)
16. Hou, Z., Hu, Z., Li, Y., Lu, R., Tang, J., Dong, Y.: Treerl: Llm reinforcement learning with on-policy tree search. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12355–12369 (2025)
17. Ji, Y., Ma, Z., Wang, Y., Chen, G., Chu, X., Wu, L.: Tree search for llm agent reinforcement learning. arXiv preprint arXiv:2509.21240 (2025)
18. Jiang, D., Guo, Z., Zhang, R., Zong, Z., Li, H., Zhuo, L., Yan, S., Heng, P.A., Li, H.: T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. arXiv preprint arXiv:2505.00703 (2025)
19. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-pic: An open dataset of user preferences for text-to-image generation. Advances in neural information processing systems **36**, 36652–36663 (2023)
20. Kong, D., Guo, Q., Xi, X., Wang, W., Wang, J., Cai, X., Zhang, S., Ye, W.: Rethinking the sampling criteria in reinforcement learning for llm reasoning: A competence-difficulty alignment perspective. arXiv preprint arXiv:2505.17652 (2025)
21. Li, H., Peng, X., Wang, Y., Peng, Z., Chen, X., Weng, R., Wang, J., Cai, X., Dai, W., Xiong, H.: Onecat: Decoder-only auto-regressive model for unified understanding and generation. arXiv preprint arXiv:2509.03498 (2025)
22. Li, J., Cui, Y., Huang, T., Ma, Y., Fan, C., Yang, M., Zhong, Z.: Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. arXiv preprint arXiv:2507.21802 (2025)
23. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. Advances in Neural Information Processing Systems **37**, 56424–56445 (2024)
24. Li, Y., Wang, Y., Zhu, Y., Zhao, Z., Lu, M., She, Q., Zhang, S.: Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. arXiv preprint arXiv:2509.06040 (2025)
25. Liang, Z., Yuan, Y., Gu, S., Chen, B., Hang, T., Li, J., Zheng, L.: Step-aware preference optimization: Aligning preference with denoising performance at each step. arXiv preprint arXiv:2406.04314 **2**(5), 7 (2024)
26. Liu, D., Zhao, S., Zhuo, L., Lin, W., Xin, Y., Li, X., Qin, Q., Qiao, Y., Li, H., Gao, P.: Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. arXiv preprint arXiv:2408.02657 (2024)
27. Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., Ouyang, W.: Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470 (2025)
28. Ma, X., Qiu, H., Zhang, G., Zeng, Z., Yang, S., Ma, L., Zhao, F.: Stage: Stable and generalizable grpo for autoregressive image generation. arXiv preprint arXiv:2509.25027 (2025)
29. Pang, Y., Jin, P., Yang, S., Lin, B., Zhu, B., Tang, Z., Chen, L., Tay, F.E., Lim, S.N., Yang, H., et al.: Next patch prediction for autoregressive visual generation. arXiv preprint arXiv:2412.15321 (2024)
30. Pang, Z., Zhang, T., Luan, F., Man, Y., Tan, H., Zhang, K., Freeman, W.T., Wang, Y.X.: Randar: Decoder-only autoregressive visual generation in random orders. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 45–55 (2025)

31. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
32. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems **36**, 53728–53741 (2023)
33. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International conference on machine learning. pp. 8821–8831. Pmlr (2021)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
35. Schuhmann, C., Beaumont, R.: Laion-aesthetics. LAION. AI **4** (2022)
36. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
37. Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., Yuan, Z.: Autoregressive model beats diffusion: Llama for scalable image generation. arXiv preprint arXiv:2406.06525 (2024)
38. Sun, Y., Shen, J., Wang, Y., Chen, T., Wang, Z., Zhou, M., Zhang, H.: Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. arXiv preprint arXiv:2506.05316 (2025)
39. Team, C.: Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint arXiv:2405.09818 (2024)
40. Tian, K., Jiang, Y., Yuan, Z., Peng, B., Wang, L.: Visual autoregressive modeling: Scalable image generation via next-scale prediction. Advances in neural information processing systems **37**, 84839–84865 (2024)
41. Tong, C., Guo, Z., Zhang, R., Shan, W., Wei, X., Xing, Z., Li, H., Heng, P.A.: Delving into rl for image generation with cot: A study on dpo vs. grpo. arXiv preprint arXiv:2505.17017 (2025)
42. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
43. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8228–8238 (2024)
44. Wang, J., Tian, Z., Wang, X., Zhang, X., Huang, W., Wu, Z., Jiang, Y.G.: Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. arXiv preprint arXiv:2504.11455 (2025)
45. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al.: Emu3: Next-token prediction is all you need. arXiv preprint arXiv:2409.18869 (2024)
46. Wang, Y., Zang, Y., Li, H., Jin, C., Wang, J.: Unified reward model for multimodal understanding and generation. arXiv preprint arXiv:2503.05236 (2025)
47. Wu, C., Chen, X., Wu, Z., Ma, Y., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C., et al.: Janus: Decoupling visual encoding for unified multimodal understanding and generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12966–12977 (2025)

48. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. arXiv preprint arXiv:2306.09341 (2023)
49. Xia, H., Ge, T., Wang, P., Chen, S.Q., Wei, F., Sui, Z.: Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 3909–3925 (2023)
50. Xie, J., Mao, W., Bai, Z., Zhang, D.J., Wang, W., Lin, K.Q., Gu, Y., Chen, Z., Yang, Z., Shou, M.Z.: Show-o: One single transformer to unify multimodal understanding and generation. arXiv preprint arXiv:2408.12528 (2024)
51. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems **36**, 15903–15935 (2023)
52. Xu, Y.E., Savani, Y., Fang, F., Kolter, J.Z.: Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. arXiv preprint arXiv:2504.13818 (2025)
53. Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al.: Dancegrpo: Unleashing grpo on visual generation. arXiv preprint arXiv:2505.07818 (2025)
54. Yang, K., Tao, J., Lyu, J., Ge, C., Chen, J., Shen, W., Zhu, X., Li, X.: Using human feedback to fine-tune diffusion models without any reward model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8941–8951 (2024)
55. You, Z., Cai, X., Gu, J., Xue, T., Dong, C.: Teaching large language models to regress accurate image quality scores using score distribution. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14483–14494 (2025)
56. Yuan, S., Liu, Y., Yue, Y., Zhang, J., Zuo, W., Wang, Q., Zhang, F., Zhou, G.: Ar-grpo: Training autoregressive image generation models via reinforcement learning. arXiv preprint arXiv:2508.06924 (2025)
57. Zhang, G., Yu, H., Ma, X., Pan, Y., Xu, H., Zhao, F.: Maskfocus: Focusing policy optimization on critical steps for masked image generation. arXiv preprint arXiv:2512.18766 (2025)
58. Zhang, G., Yu, H., Ma, X., Zhang, J., Pan, Y., Yao, M., Xiao, J., Huang, L., Zhao, F.: Group critical-token policy optimization for autoregressive image generation. arXiv preprint arXiv:2509.22485 (2025)
59. Zhang, H., Qu, L., Liu, Y., Chen, H., Song, Y., Dong, Y., Sun, S., Li, X., Wang, X., Jiang, Y., et al.: Nextflow: Unified sequential modeling activates multimodal understanding and generation. arXiv preprint arXiv:2601.02204 (2026)
60. Zhang, R., Arora, D., Mei, S., Zanette, A.: Speed-rl: Faster training of reasoning models via online curriculum learning. arXiv preprint arXiv:2506.09016 (2025)
61. Zhang, Y., Lv, N., Wang, T., Dang, J.: Fastgrpo: Accelerating policy optimization via concurrency-aware speculative decoding and online draft learning. arXiv preprint arXiv:2509.21792 (2025)
62. Zhang, Y., Yao, W., Yu, C., Liu, Y., Yin, Q., Yin, B., Yun, H., Li, L.: Improving sampling efficiency in rlvr through adaptive rollout and response reuse. arXiv preprint arXiv:2509.25808 (2025)
63. Zheng, H., Zhou, Y., Bartoldson, B.R., Kailkhura, B., Lai, F., Zhao, J., Chen, B.: Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. arXiv preprint arXiv:2506.02177 (2025)

64. Zheng, K., Chen, H., Ye, H., Wang, H., Zhang, Q., Jiang, K., Su, H., Ermon, S., Zhu, J., Liu, M.Y.: Diffusionnft: Online diffusion reinforcement with forward process. arXiv preprint arXiv:2509.16117 (2025)
65. Zhou, C., Yu, L., Babu, A., Tirumala, K., Yasunaga, M., Shamis, L., Kahn, J., Ma, X., Zettlemoyer, L., Levy, O.: Transfusion: Predict the next token and diffuse images with one multi-modal model. arXiv preprint arXiv:2408.11039 (2024)