RankE: End-to-End Post-Training for Discrete Text-to-Image Generation with Decoder Co-Evolution

Siyong Jian[⋆1] **Siyuan Li**[⋆1,2] **Luyuan Zhang**[⋆3] **Zedong Wang**[4]
Xin Jin[1] **Ying Li**[1] **Cheng Tan**[†5] **Huan Wang**[†1]

[1]Westlake University [2]Zhejiang University [3]Tsinghua University
[4]Hong Kong University of Science and Technology [5]Shanghai AI Lab

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

Project page: `https://syjmelody.github.io/RankE/`

Abstract

Discrete autoregressive (AR) text-to-image (T2I) models pair a VQ tokenizer with an AR policy. Current post-training methods optimize the policy while keeping the VQ decoder frozen. We identify *Latent Covariate Shift*: reward-driven policy updates change the token distribution relative to the ground-truth codes on which the decoder was trained, so alignment can improve while decoded image fidelity degrades. We propose *RankE*, an end-to-end post-training framework that co-evolves the policy and decoder through alternating optimization. Group-relative policy optimization refines the policy, while reward-aware adversarial training, differentiable reward feedback, reconstruction, and EMA consistency adapt the decoder. On LlamaGen-XL (775M), RankE improves both FID and CLIP on MS-COCO 30K (15.21 and 33.76), whereas frozen-decoder RL improves CLIP but worsens FID relative to SFT. Paired runs, decoder interventions, and an approximately wall-clock-matched checkpoint comparison strengthen this result. Janus-Pro (1B) shows more modest gains over frozen-decoder RL under matched post-training conditions; it does not recover the Base model's fidelity after SFT. These results support decoder co-evolution as a way to improve the fidelity–alignment operating point, with the strongest evidence on LlamaGen-XL.

1 Introduction

Discrete autoregressive (AR) text-to-image (T2I) models factorize image generation into two stages: a VQ tokenizer [57, 14] maps images to discrete codebook entries, and an AR policy models the resulting token sequences via next-token prediction [26, 54]. This formulation enables unified multimodal architectures [5, 10] and directly inherits the favorable scaling behavior and infrastructure of large language models. The alignment of these models increasingly relies on post-training [59, 69, 24], which conventionally optimizes only the AR policy while keeping the VQ decoder frozen.

This frozen-decoder convention is increasingly out of step with recent progress on the continuous T2I side. Diffusion methods such as REPA-E [29] have begun to unlock the VAE for joint optimization with the denoiser, thereby lifting a frozen-decoder assumption that has long been treated as a default in latent generative modeling. In discrete AR, the picture is precisely the opposite: existing post-training methods [24, 59, 69, 32] typically freeze the VQ decoder and optimize only the AR policy.

We identify the underlying mismatch as *Latent Covariate Shift* (Fig. 2 (a)). During tokenizer pre-training, the VQ decoder is trained exclusively on deterministic ground-truth codes $z_{\mathrm{gt}} = \mathrm{Quantize}(E(x))$ [57, 14], which occupy a restricted, low-variance region of the latent space [48]. At inference, however, the same decoder receives tokens sampled from the AR policy, $\hat{z} \sim \pi_\theta(\cdot \mid y)$, whose distribution can diverge from the decoder's training regime under reward-driven policy updates.

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

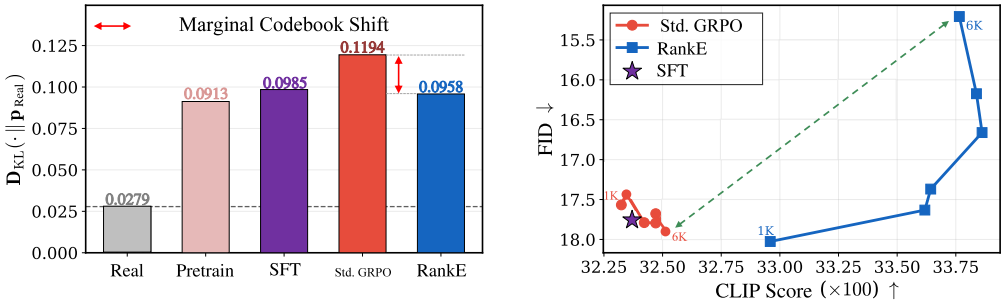

Figure 1: **Marginal codebook shift and fidelity–alignment trajectories on LlamaGen-XL. Left:** KL divergence between each model's global marginal codebook frequencies and that of 5,000 real MS-COCO images; the dashed line marks *Real*, a finite-sample variation reference computed from two independently sampled real-image sets encoded by the same frozen tokenizer. Std. GRPO widens the endpoint marginal gap by 21% over SFT, while RankE returns it near the SFT level; per-step dynamics are shown in Fig. 6. **Right:** CLIP–FID trajectory across training checkpoints. Std. GRPO shows CLIP gains with stagnant or degrading FID; RankE shows an overall trend toward higher CLIP and lower FID over the plotted checkpoints. These are representative single-run observations; marginal KL alone does not establish causality or capture spatial, co-occurrence, or prompt-conditional shift.

This divergence can contribute to a fidelity–alignment trade-off: GRPO [51] applied to LlamaGen-XL [54] improves CLIP while FID stagnates or degrades over the evaluated checkpoints (Fig. 1, right). The global marginal KL diagnostic (Fig. 1, left) also shows a larger endpoint gap under Std. GRPO than under SFT; it does not by itself establish that the shift causes the fidelity change. Unlike exposure bias [1, 47], which concerns the input context of the generator, Latent Covariate Shift targets the input distribution of the decoder—a mismatch that policy-only reward optimization does not explicitly address.

Adapting the decoder alongside the policy directly addresses this mismatch, but direct end-to-end optimization is blocked by two non-differentiable operations along the generation chain: categorical sampling at $z \sim \pi_\theta$ and VQ quantization [23, 2]. Together, these operations sever the gradient path from pixel-space rewards to policy parameters (Fig. 2 (a))—a barrier that simply does not arise in continuous diffusion models, where the generation chain remains fully differentiable [11, 44, 29]. Standard surrogates [2, 21, 23] introduce non-trivial gradient bias or training instability at the codebook scales used by modern visual tokenizers [25].

We introduce **RankE** (**Rank**ing-based **E**nd-to-end alignment), a post-training framework that, to our knowledge, is the first to co-evolve the policy and VQ decoder in discrete AR T2I post-training without differentiating through the discrete bottleneck. The name reflects two ranking-based mechanisms that operate at complementary granularities: a *token-level* ranking objective (group-relative advantages in GRPO) drives the policy update, and a *pixel-level* ranking objective (a reward-weighted adversarial loss, *Rank-GAN*) drives the decoder update. As illustrated in Fig. 2 (b), RankE employs an alternating optimization strategy that admits an approximate GEM-inspired block-coordinate interpretation (Sec. 3.2). In the *policy stage*, the AR generator is updated via group-relative preference optimization [51] with KL regularization. In the *decoder stage*, the VQ decoder is adapted on policy-sampled latents through Rank-GAN and EMA-anchored consistency regularization, which together help constrain drift from its reconstruction prior.

By allowing the decoder to track the policy's evolving token distribution, RankE directly targets Latent Covariate Shift and improves the observed fidelity–alignment operating point (Fig. 1, right): on LlamaGen-XL (775M), it improves FID to 15.21 and CLIP to 33.76 on MS-COCO 30K, whereas Std. GRPO improves alignment at the expense of fidelity. On Janus-Pro-1B and under the HPSv2 reward, RankE improves alignment (CLIP/HPSv2) and GenEval over the Std. GRPO baseline in the reported single runs, supporting improvement over frozen-decoder RL under matched post-training conditions, with more modest gains on Janus-Pro.

Our contributions are summarized as follows:

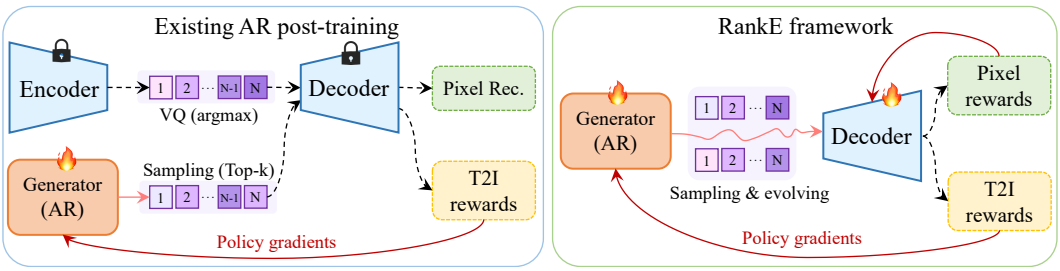

Figure 2: Comparison of existing AR post-training and the RankE framework. **Left:** After pre-training the VQ tokenizer, existing works [10, 59] only optimize the AR generator with the sampled latent rollouts via the RL ranking loss (policy gradients) with T2I rewards while freezing the decoder, leaving the decoder unadapted to changes in policy-sampled tokens. **Right:** RankE applies an alternating training pipeline to co-evolve both the generator and the decoder via RL objectives with T2I rewards (for the generator & decoder) and pixel rewards (for the decoder). The decoder stage adapts the renderer to policy-sampled latents while retaining reconstruction and consistency anchors. Fixed-latent decoder interventions support this decoder-mismatch interpretation (Appendix I.4).

- We identify *Latent Covariate Shift*—a decoder-side distribution mismatch distinct from generator-side exposure bias—and observe increased marginal codebook shift under policy-only GRPO in the evaluated setting.
- We develop alternating policy and decoder updates that combine GRPO with reward-aware decoder adaptation and consistency regularization, without differentiating through sampled token indices.
- We demonstrate simultaneous fidelity and alignment gains on LlamaGen-XL, supported by two paired runs, decoder interventions, and an approximately time-matched comparison. Janus-Pro and higher-resolution experiments provide additional evidence of improvement over frozen-decoder RL under matched post-training conditions.

2 Related Work

Post-Training and Alignment in T2I Post-training for T2I has matured rapidly in the diffusion family. Online RL [4, 15], offline preference optimization [58], and direct reward fine-tuning [65, 11, 44] all exploit a key structural property that is unavailable in discrete AR: the denoising chain is differentiable end-to-end, so reward gradients can flow from pixel space back to the generator via $\nabla_\theta R(\mathbf{x})$. More recently, REPA-E [29] goes one step further by unlocking the VAE for joint optimization with the denoiser, lifting the frozen-decoder assumption that has long been treated as a default in latent generative modeling, and demonstrating that decoder adaptation yields gains in both fidelity and alignment. For discrete AR, by contrast, post-training is far less developed. Methods such as T2I-R1 [24], SimpleAR [59], GCPO [69], and VA-π [32] apply GRPO [51] to the AR policy. These methods keep the VQ decoder frozen: reward is computed in pixel space, yet only the policy is updated. A complementary line of work [30, 28] reframes KL-regularized RL as variational inference under a reward-induced log-likelihood; we use this perspective to give RankE's alternation an approximate GEM-inspired block-coordinate interpretation (§3.2). RankE instead adapts the decoder alongside the policy during discrete AR post-training, allowing the renderer to track changes in policy-sampled latents (§3). A broader survey of discrete visual tokenizers, AR generator factorizations, and the gradient barrier is deferred to Appendix A.

3 Method

3.1 Problem Formulation

A discrete AR T2I system consists of an autoregressive token policy $\pi_\theta(z \mid y)$ and a VQ decoder D_ϕ that renders codes into images via $x = D_\phi(z)$. Given a reward function r that scores text–image alignment or human preference [65, 63], we seek to maximize

$$\max_{\theta,\phi} \mathbb{E}_{y\sim\mathcal{D},\, z\sim\pi_\theta(\cdot|y)}[\, r(D_\phi(z),\, y)\,]\,. \tag{1}$$

This single objective ties both modules to one pixel-space reward, yet categorical sampling at $z \sim \pi_\theta$ and VQ quantization [23, 2] sever the gradient path: signals flow into the decoder but cannot reach the policy. Rather than forcing this bottleneck with a biased surrogate, RankE alternates *around* it—each module is updated with the signal natural to its own parameter space, and reward information crosses the gap through the interleaving of the two updates. Sec. 3.2 casts this alternation as a unified regularized objective and gives an approximate GEM-inspired block-coordinate interpretation.

3.2 Alternating Co-Evolution Around the Discrete Bottleneck

A unified two-stage objective. The updated parameter $\Psi \in \{\theta, \phi\}$ maximizes a regularized alignment objective

$$\max_{\Psi \in \{\theta,\phi\}} \mathcal{J}(\Psi) = \underbrace{\mathbb{E}[\mathcal{A}_\Psi]}_{\text{ranking-based alignment}} - \underbrace{\lambda\,\Omega(\Psi)}_{\text{stability-preserving regularization}}, \tag{2}$$

where $\mathcal{A}_\Psi$ pushes Ψ toward the reward-favored region and $\Omega(\Psi)$ keeps it tethered to a trusted prior. The policy uses group-relative advantages, while the decoder combines direct reward gradients with reward-weighted realism and stability terms. These complementary updates share reward information through alternation over K rounds.

Stage 1: token-level ranking via GRPO. With the decoder fixed, we update the policy using Group Relative Policy Optimization [51], which converts reward scalars into a per-prompt ranking at the token level. For each prompt y, we draw G rollouts $\{z_i\}_{i=1}^G \sim \pi_\theta(\cdot \mid y)$, decode them with the frozen D_ϕ, score them under r, and form group-normalized advantages $A_i = (r_i - \mu_r)/\sigma_r$ [61]. For a constant-reward group ($\sigma_r = 0$), we set $A_i = 0$. The resulting loss

$$\mathcal{L}_\pi(\theta) = -\,\mathbb{E}_y\left[\underbrace{\frac{1}{G}\sum_{i=1}^{G}\min\big(\rho_i A_i,\ \mathrm{clip}(\rho_i, 1\pm\epsilon)\,A_i\big)}_{\mathcal{A}_\theta\text{: token-level ranking}} - \underbrace{\beta\,\mathbb{D}_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big)}_{\Omega_\theta\text{: stability anchor}}\right], \tag{3}$$

maps cleanly onto Eq. 2: the clipped advantage term is the token-level ranking signal $\mathcal{A}_\theta$, and the KL against an EMA reference [16, 12] serves as the stability anchor Ω_θ. Here, $\rho_i = \pi_\theta(z_i \mid y)/\pi_{\theta_{\mathrm{old}}}(z_i \mid y)$ denotes the PPO importance ratio [50].

Stage 2: pixel-level ranking, at a glance. With the policy fixed, we allow the decoder to track its evolving token distribution. Fresh groups of G detached rollouts are sampled from the frozen policy snapshot at the end of Stage 1: decoded images preferred by the reward model receive a larger weight in the decoder update, less-preferred samples are down-weighted, and the gradient pulls D_ϕ toward outputs resembling the top-ranked decodings. Mirroring the structure of Stage 1, the decoder loss decomposes into a ranking-based alignment block and a manifold-anchored regularization block:

$$\mathcal{L}_D(\phi) = \underbrace{\lambda_d\,\mathcal{L}_{\text{reward}} + \lambda_g\,\mathcal{L}_{\text{Rank-GAN}}}_{\mathcal{A}_\phi\text{: pixel-level ranking}} + \underbrace{\lambda_r\,\mathcal{L}_{\text{recon}} + \lambda_c\,\mathcal{L}_{\text{consist}}}_{\Omega_\phi\text{: manifold anchor}}. \tag{4}$$

The two stages share a regularized reward-guided design, but their gradients differ: $\mathcal{A}_\phi$ ranks policy-sampled decodings in pixel space via a reward-weighted adversarial signal (*Rank-GAN*), while Ω_ϕ anchors the decoder to the deterministic ground-truth manifold on which the tokenizer was trained.

Why alternation, and a GEM view. RankE alternates n_π policy-only steps (decoder frozen) and n_D decoder-only steps (policy frozen) for K rounds. Separate AdamW optimizers update the policy, decoder, and discriminator. This separates stochastic policy gradients from differentiable decoder updates without differentiating through sampled token indices, allowing each module to use its appropriate gradient estimator. RankE admits an *approximate GEM-inspired block-coordinate interpretation* [30, 28]: an idealized policy step improves a variational reward objective, and an idealized decoder step improves that objective with fixed regularization. The practical clipped GRPO, finite-sample advantages, adversarial loss, and moving EMA references do not guarantee monotonic ELBO improvement or convergence of every stochastic update. Appendix B distinguishes the idealized objective from the implemented procedure; Algorithm 1 specifies the alternating updates.

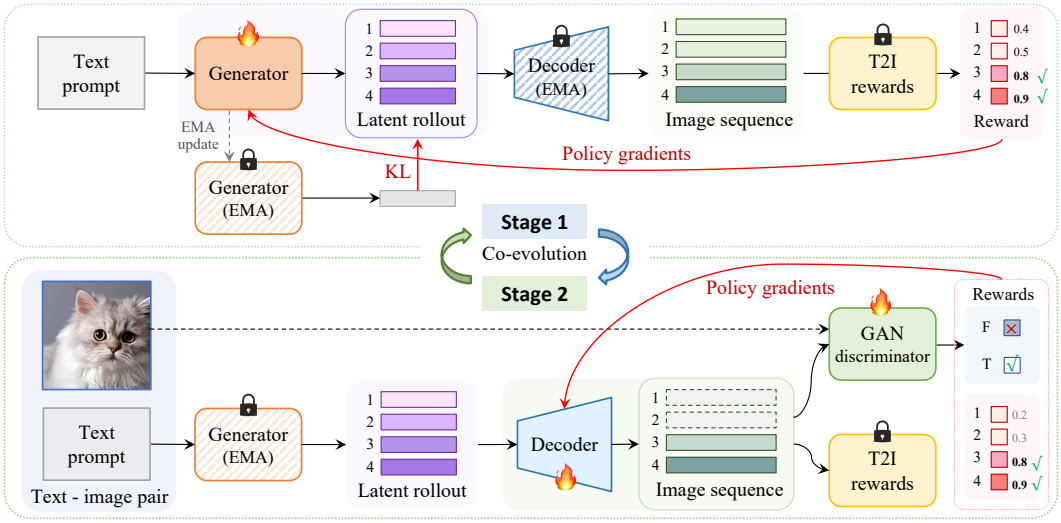

Figure 3: Overview of the RankE co-evolution framework. **Stage 1 (Policy Alignment):** The AR policy is updated via policy-gradient optimization to maximize a human-preference reward, with KL regularization toward an EMA reference policy. The VQ decoder is frozen; this stage targets alignment and can change the decoder-input distribution. **Stage 2 (Decoder Adaptation):** The policy is frozen and the VQ decoder is updated on policy-sampled latents via reward-weighted adversarial supervision (Rank-GAN), differentiable reward gradients, and EMA consistency regularization. Alternating the two stages couples policy improvement with decoder adaptation, adapting the renderer to the changing decoder-input distribution.

3.3 Decoder Adaptation: Reward-Driven Alignment Meets Manifold Anchoring

3.3.1 Reward-driven alignment ($\mathcal{A}_\phi$)

Because T2I rewards come in two flavors—differentiable scorers such as CLIP and scorers used as black boxes—we adopt two complementary channels rather than one, with the choice dictated by the reward.

Differentiable channel: direct reward back-propagation. When the reward R admits gradients, the decoder offers a fully differentiable path from latents to scalar feedback. Following differentiable reward fine-tuning for diffusion [11, 44], we maximize R through the decoder:

$$\mathcal{L}_{\text{reward}}(\phi) = -\,\mathbb{E}_{\hat{z}\sim\pi_\theta}\big[\,R(D_\phi(\hat{z}),\,y)\,\big]\,. \tag{5}$$

Crucially, $\hat{z}$ is policy-sampled and *detached*: no gradient crosses the discrete boundary, so this channel never attempts the impossible task of differentiating through categorical sampling.

Black-box channel: Rank-GAN. When R is non-differentiable, the channel above vanishes. A vanilla GAN loss [18] on policy-sampled latents would treat every rollout uniformly and discard the per-sample ranking that the policy has just been optimized over. We therefore introduce a reward-weighted variant inspired by reward-weighted regression [42, 41], which we call *Rank-GAN*:

$$\mathcal{L}_{\text{Rank-GAN}}(\phi) = -\,\mathbb{E}_{\hat{z}\sim\pi_\theta}\big[\,w(\hat{z})\cdot\text{Disc}(D_\phi(\hat{z}))\,\big]\,, \tag{6}$$

with detached weights $w_i = \text{sg}[G\exp(r_i/\tau)/\sum_{j=1}^{G}\exp(r_j/\tau)]$, so $\sum_i w_i = G$. The unit mean of the weights controls their overall scale but does not guarantee equal gradient norms. Rank-GAN provides *reward-conditioned realism adaptation*, not a separate semantic target: it emphasizes the rendering of reward-preferred latents. The discriminator is trained adversarially against real images x_{gt}. The step- and schedule-matched 2×2 ablation in Appendix I.5 separates this effect from direct reward backpropagation, which supplies the dominant semantic-alignment gradient.

What the two channels share. Direct reward backpropagation supplies semantic gradients when enabled; detached reward weighting allocates adversarial realism gradients across samples. Rank-GAN can use scalar rewards without differentiating through the scorer. For a scorer used only as

a black box, $\lambda_d = 0$; this is an implementation choice, not a claim that the HPSv2 architecture is intrinsically non-differentiable. Under CLIP, both channels are used. Their distinct contributions are measured in Appendix I.5.

3.3.2 Manifold-anchored regularization (Ω_ϕ)

Alignment alone is unsafe: trained only on stochastic policy latents under adversarial pressure, the decoder would readily abandon the deterministic ground-truth manifold on which it was originally fit—the renderer-side analogue of reward hacking. Two regularizers prevent this drift, each targeting a distinct failure mode.

Anchor 1: reconstruction on ground-truth codes. We retain the original tokenizer-training objective on ground-truth codes z_{gt} [14]:

$$\mathcal{L}_{\text{recon}}(\phi) = \left\| x_{\text{gt}} - D_\phi(z_{\text{gt}}) \right\|_1 + \mathcal{L}_{\text{GAN}}(D_\phi(z_{\text{gt}})) \,. \tag{7}$$

Mixed into every M-step, this term preserves fidelity on the deterministic ground-truth distribution against catastrophic forgetting [27] induced by training on stochastic policy samples.

Anchor 2: EMA-consistent stability on policy codes. A subtler concern is local stability. A single token-index change can produce a large pixel change; the consistency penalty discourages abrupt changes in rendering during adaptation. To smooth its response in novel code regions, we distill from a slow-moving EMA teacher $D_{\phi_{\text{ema}}}$ [70], following the self-distillation paradigm [55, 19]:

$$\mathcal{L}_{\text{consist}}(\phi) = \mathbb{E}_{\hat{z}\sim\pi_\theta}\left[\, \mathcal{L}_{\text{LPIPS}}\big(D_\phi(\hat{z}),\ \text{sg}[D_{\phi_{\text{ema}}}(\hat{z})] \big) \,\right] . \tag{8}$$

The teacher provides a stable target that filters out the high-frequency noise of single-step adversarial updates on a discrete input. The two anchors are complementary—$\mathcal{L}_{\text{recon}}$ guards against manifold forgetting on ground-truth codes, while $\mathcal{L}_{\text{consist}}$ guards against over-fitting to whatever the policy happens to sample on a given step—and together they regularize decoder drift while allowing reward-guided adaptation.

4 Experiments

Our experiments address a core question: under matched reward, data, and policy-update budgets, does co-evolving the VQ decoder with the AR policy yield measurable gains over the frozen-decoder convention? We establish a controlled setting (Sec. 4.1), conduct three tests (Sec. 4.2), investigate the proposed mechanism (Sec. 4.3), and isolate component contributions (Sec. 4.4).

4.1 Experimental Setup

Models and baselines. We evaluate RankE on two representative discrete AR T2I backbones: LlamaGen-XL [54] (775M) and Janus-Pro-1B [10], a unified multimodal architecture. We compare against three baselines of increasing post-training intensity: the pre-trained model (Base), a supervised fine-tuned variant on our curated corpus (SFT), and a standard GRPO baseline that updates the AR policy via GRPO [51] under a CLIP [45] or HPSv2 [63] reward while keeping the decoder frozen (Std. GRPO). The main comparison shares initialization, reward, data, and policy-update budget. Decoder adaptation incurs additional compute, so we separately report an approximately wall-clock/GPU-hour-matched comparison (Appendix I.2). Neither comparison is strictly FLOP-matched.

Datasets and evaluation. We evaluate along three complementary axes: fidelity and alignment through FID [20] and CLIP Score [45] on MS-COCO 30K [33]; automated human-preference scoring through HPSv2 [63] across all four subsets; and compositional performance on GenEval [17]. We show selected categories (Two-Object, Counting, and Colors) alongside the overall score across all six benchmark tasks. Because the 15K training corpus includes 750 GenEval-sourced examples, we report these scores as benchmark performance rather than zero-shot generalization (Appendix H). The 15K training corpus is curated from BLIP3o-60k with caption compression and stratified domain sampling, and full details are given in Appendix H.

4.2 Does Decoder Co-Evolution Help? Three Controlled Tests

We test our central question with three lenses, moving from the broadest to the most stringent.

Table 1: Comparison of representative T2I alignment methods. We focus on whether the decoder (VAE or VQ) is updated during post-training (**Dec.**). Most methods keep the decoder frozen; RankE is, to our knowledge, the first discrete AR method to co-evolve the decoder. FID and CLIP are reported on MS-COCO 30K where available; "–" indicates the metric is not reported by the original work under a comparable protocol. Numbers from prior work are cited as-is and may use different reward signals or evaluation pipelines, thus not apples-to-apples; more comparison appears in Tab. 2.

Method	Venue	Size	Type	Strategy	Dec.	FID↓	CLIP↑
Diffusion & Flow Matching							
SDXL [43]	arXiv'23	2.6B	Diff.	Pretrain	Frozen	–	31.0
DPOK [15]	NeurIPS'23	0.9B	Diff.	RL + Direct BP	Frozen	–	–
Diffusion-DPO [58]	CVPR'24	2.6B	Diff.	DPO	Frozen	10.5	31.5
REPA-E [29]	ICCV'25	LoRA	Diff.	E2E Reparam.	Unfrozen	–	–
Flow-GRPO [35]	NeurIPS'25	2.0B	Flow	Online GRPO	Frozen	10.5	–
Autoregressive (AR) & Unified							
LlamaGen [54]	arXiv'24	775M	AR	Pretrain	Unfrozen	15.24	31.54
Transfusion [71]	ICLR'25	7.0B	Unified	Mixed-Modal	Frozen	16.9	25.5
Emu3 [60]	arXiv'24	7.0B	AR	Unified AR	Frozen	11.6	28.6
Janus-Pro [10]	arXiv'25	7.0B‡	Uni.	Decoupled Enc.	Frozen	6.85	32.0
GCPO [69]	ICLR'26	775M	AR	Critical-Token Opt.	Frozen	–	–
VA-π [32]	CVPR'26	775M	AR	Variational Align.§	Frozen	9.23	29.1
LlamaGen-SFT	–	775M	AR	SFT	Frozen	16.58	31.86
LlamaGen-RL	–	775M	AR	GRPO	Frozen	17.76	32.45
RankE (Ours)	–	**775M**	**AR**	**Co-Evolved**	**Unfrozen**	**15.21**	**33.76**

‡The 7B Janus-Pro number is included for reference; our controlled experiments use Janus-Pro-1B. §VA-π reports FID under a different evaluation protocol (see [32] for details), so its FID is not directly comparable to ours.

Table 2: Quantitative results under CLIP-based optimization. On LlamaGen-XL, Std. GRPO improves CLIP but worsens FID relative to SFT; on Janus-Pro, it improves both relative to SFT but does not recover Base FID. RankE achieves higher CLIP and lower FID than Std. GRPO on both backbones in these representative runs. GenEval shows three selected categories and the overall score across all six tasks; the training corpus contains 750 GenEval-sourced examples. Green numbers denote gains over Std. GRPO.

Backbone	Method (Reward)	Decoder	MS-COCO		GenEval			
			CLIP↑	FID↓	2-Obj	Count	Colors	Overall↑
LlamaGen-XL (775M)	Base	Frozen	31.54	15.24	0.331	0.175	0.559	0.309
	SFT	Frozen	31.86	16.58	0.288	0.228	0.723	0.374
	Std. GRPO (*CLIP*)	Frozen	32.45	17.76	0.361	0.272	0.779	0.417
	RankE (*CLIP*)	**Co-Evol**	**33.76** (↑1.31)	**15.21** (↓2.55)	0.379	0.356	0.763	**0.425** (↑.008)
Janus-Pro (1B)	Base	Frozen	33.20	18.95	0.810	0.553	0.894	0.740
	SFT	Frozen	33.31	26.73	0.826	0.572	0.889	0.739
	Std. GRPO (*CLIP*)	Frozen	33.60	25.59	0.838	0.478	0.891	0.746
	RankE (*CLIP*)	**Co-Evol**	**33.86** (↑0.26)	**25.19** (↓0.40)	0.836	0.487	0.892	**0.750** (↑.004)

Test 1: positioning across the post-training landscape. Table 1 situates RankE within the broader post-training picture, spanning diffusion models [49, 40], flow matching [34], unified architectures [56], and AR generators [67]. Among methods that explicitly target the CLIP/FID alignment trade-off on MS-COCO 30K, RankE attains a favorable operating point of FID 15.21 and CLIP 33.76 at 775M parameters; comparisons that report different reward signals or evaluation protocols are not directly commensurable.

Test 2: apples-to-apples controlled comparison. Under a CLIP reward (Tab. 2), Std. GRPO improves CLIP but degrades FID on LlamaGen-XL (16.58 → 17.76). This pattern is consistent with a mismatch between policy-sampled latents and the frozen decoder's training distribution. RankE reverses this, reaching FID 15.21 and CLIP 33.76. On Janus-Pro-1B, RankE yields the best

Table 3: Quantitative results under HPSv2-based optimization (representative single run). RankE exceeds Std. GRPO in HPSv2 average and GenEval overall. HPSv2 shows three categories; its average includes all four, including Paintings. GenEval shows selected categories and the six-task overall score. As training includes GenEval-sourced samples, these scores measure benchmark performance, not zero-shot generalization.

Backbone	Method (Reward)	Decoder	HPSv2				GenEval			
			Photo	Concept	Anime	**Avg.** ↑	2-Obj	Count	Colors	**Overall** ↑
	Base	Frozen	0.2364	0.2107	0.2183	0.2196	0.331	0.175	0.559	0.309
LlamaGen-XL (775M)	SFT	Frozen	0.2281	0.2202	0.2222	0.2221	0.288	0.228	0.723	0.374
	Std. GRPO (*HPSv2*)	Frozen	0.2466	0.2435	0.2436	0.2451	0.397	0.259	0.798	0.418
	RankE (*HPSv2*)	**Co-Evol**	**0.2492**	**0.2479**	**0.2453**	**0.2531**	0.386	**0.291**	0.771	**0.423**

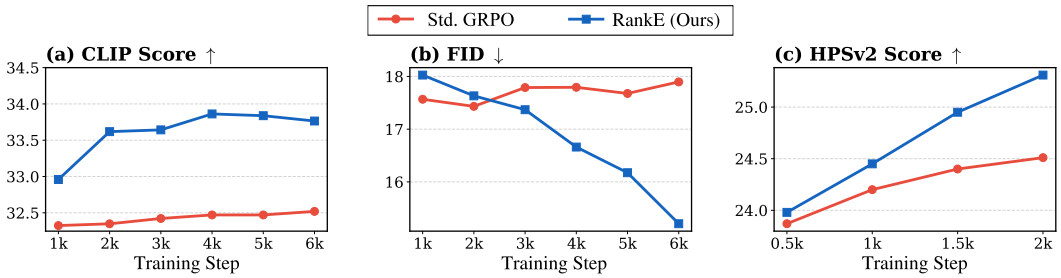

Figure 4: **Evolution of generation metrics over training steps.** Comparison of RankE against a Std. GRPO baseline with a frozen decoder. Across the plotted checkpoints, Std. GRPO shows alignment gains with stagnant or degrading fidelity (b). RankE shows overall gains in alignment (a), fidelity (b), and human preference scores (c). These representative single-run trajectories are descriptive evidence; decoder interventions are reported separately in Appendix I.4.

CLIP and GenEval; all post-training variants here regress FID vs. Base, likely due to a corpus-matching limitation (Sec. 5). A Base-initialized Janus-Pro comparison gives CLIP/FID 33.50/19.30 for Std. GRPO and 33.65/19.00 for RankE, compared with Base 33.20/18.95 (Appendix I.3). This isolates the preceding SFT effect but is a single-run result, not evidence of Base-fidelity recovery. With HPSv2 as scalar feedback (Tab. 3), RankE raises the average score from 0.2451 to 0.2531 while maintaining the GenEval average in this run. This supports evaluation under a second reward signal; it does not isolate decoder adaptation from reward-specific effects.

Test 3: qualitative verification. Fig. 5 compares generations under matched prompts. The base model frequently misses prompt attributes (color, count, spatial relations). Std. GRPO improves adherence but also shows visible artifacts; this observation is consistent with the frozen decoder receiving policy-shifted latents. RankE produces images with faithful attributes and fewer visible artifact bands. These comparisons suggest that decoder adaptation can help translate latent-space alignment into pixel-space fidelity.

4.3 Why Does It Work? Diagnosing the Mechanism

Our KL diagnostic measures *global marginal codebook usage*, not a sequence-level or prompt-conditional policy KL. With $N = 5{,}000$ samples and L tokens each, $p_\theta(c) = (NL)^{-1} \sum_{n,\ell} \mathbf{1}[z_{n\ell} = c]$; p_{GT} is computed from real images encoded by the same frozen tokenizer. We report $D_{\mathrm{KL}}(p_\theta \| p_{\mathrm{GT}})$. The marginal drift and entropy curves in Fig. 6 are useful aggregate diagnostics, but do not establish causality on their own.

Complementary interventions strengthen the decoder-mismatch interpretation (Appendix I.4). Position-wise JS increases from 0.0314 (SFT) to 0.0468 (Std. GRPO), and is 0.0459 under RankE despite better FID. On identical late-RL latents, replacing only the decoder improves FID from 17.63 to 15.28 and CLIP from 33.32 to 33.71. Controlled nearest-neighbor token replacements increase both marginal KL and FID while only slightly reducing CLIP. Together, these observations support decoder adaptation as a contributor to the gains, while neither marginal KL nor the interventions establish it as the sole causal mechanism.

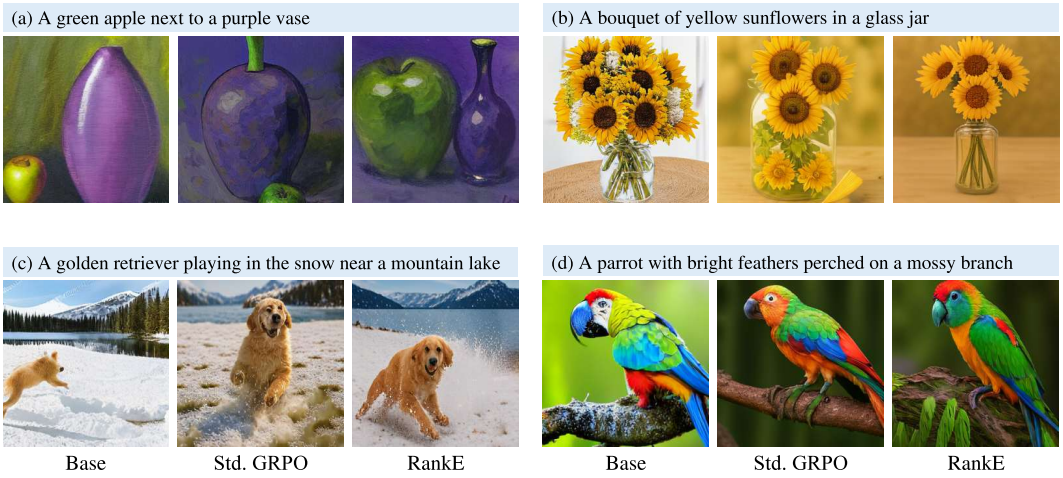

Figure 5: Visualization of T2I generation. RankE yields precise attributes and details according to the text prompt compared to the baseline, while reducing artifacts with high image quality compared to the existing RL method (GRPO).

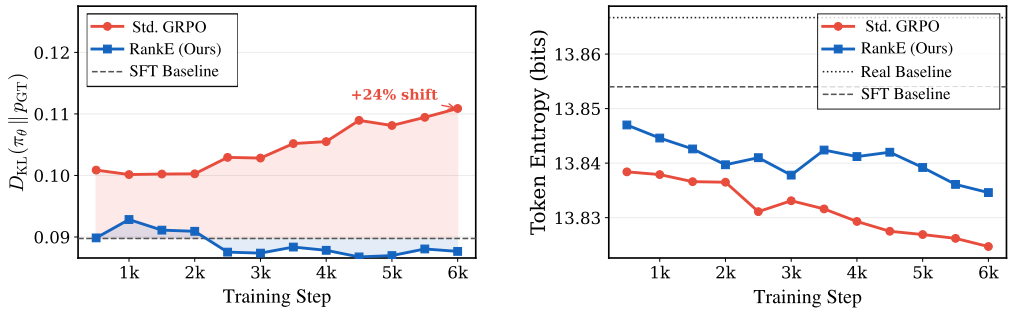

Figure 6: Latent Covariate Shift and token entropy during training. **Left:** global marginal codebook shift measured as $D_{\mathrm{KL}}(p_\theta \,\|\, p_{\mathrm{GT}})$, evaluated every 500 steps. Std. GRPO with a frozen decoder shows steadily increasing divergence ($+24\%$ over training), whereas RankE keeps the divergence near the SFT initialization. **Right:** Std. GRPO reduces token entropy as the policy concentrates on fewer codebook entries; RankE maintains entropy closer to the real-image level (13.87 bits). These are representative single-run diagnostics; they omit spatial, co-occurrence, and prompt-conditional information and do not establish causality.

4.4 What Drives the Gains? Component-Level Ablation

Having compared co-evolution with single-module variants and examined evidence consistent with the proposed mechanism, we isolate the contributions of its design choices. We ablate two design axes on LlamaGen-XL under a CLIP reward on MS-COCO 30K: the high-level training mode and the composition of the decoder loss. Further sensitivity analyses appear in Appendix E.

Training mode: is co-evolution synergistic, or just the union of two effects? Tab. 4 compares four configurations. Policy-only GRPO (Row 2) improves CLIP but degrades FID, which is the Latent Covariate Shift diagnosis again. Decoder-only adaptation (Row 3) raises CLIP to 33.41 without a policy-level update, suggesting a decoder-side opportunity under this setup; FID nonetheless degrades to 18.68 without coordinated policy guidance. Only full RankE (Row 4) is best on both fidelity (FID) and overall composition (GenEval), and achieves the best CLIP. On FID, full RankE (15.21) improves over either policy-only (17.76) or decoder-only (18.68), suggesting complementary contributions from the two updates under this configuration.

Decoder loss ablation: which term inside $\mathcal{L}_D$ matters? Tab. 5 removes each term individually, which separates the alignment block from the regularization block of Eq. 4. Removing reconstruction

Table 4: Effect of training mode (CLIP reward, MS-COCO 30K). Full joint training outperforms either single-module configuration in this run.

Setting	Policy	Decoder	CLIP↑	FID↓	GenEval↑
SFT (No Post-Train)	✗	✗	31.86	16.58	0.374
Policy-Only (GRPO)	✓	✗	32.45	17.76	0.417
Decoder-Only	✗	✓	33.41	18.68	0.403
Full RankE	✓	✓	**33.76**	**15.21**	**0.425**

Table 5: Contribution of each decoder loss component under joint training (CLIP reward, MS-COCO 30K). Each term is removed individually from the full configuration to measure its marginal effect.

Setting	$\mathcal{L}_r$	$\mathcal{L}_g$	$\mathcal{L}_c$	$\mathcal{L}_d$	CLIP↑	FID↓	GenEval↑
w/o Reconstruction	✗	✓	✓	✓	33.26	17.69	0.401
w/o GAN	✓	✗	✓	✓	32.12	18.59	0.397
w/o Consistency	✓	✓	✗	✓	34.17	19.03	0.408
w/o Reward BP	✓	✓	✓	✗	32.53	20.16	0.412
Full RankE	✓	✓	✓	✓	**33.76**	**15.21**	**0.425**

worsens FID to 17.69, while removing consistency raises CLIP to 34.17 but worsens FID to 19.03, illustrating the need to constrain decoder drift. The factorial ablation (Appendix I.5) shows that direct reward backpropagation supplies the dominant alignment gain. With reward backpropagation enabled, Rank-GAN improves CLIP/FID from $33.48/15.92$ (uniform GAN) to $33.76/15.21$. Its role is reward-conditioned realism adaptation; the additional CLIP gain is an indirect rendering benefit.

Statistical reliability and compute budget. Across two paired LlamaGen-XL training runs, Std. GRPO obtains CLIP 32.41 ± 0.06 and FID 17.85 ± 0.12, while RankE obtains 33.67 ± 0.13 and 15.37 ± 0.22 (mean $\pm$ sample standard deviation). Both pairs improve both metrics; two runs are a reproducibility check rather than a significance test (Appendix I.1). At 6K policy updates, RankE adds 3.0% peak memory and 7.9% wall-clock time. Its 5.5K checkpoint uses approximately 150.3 GPU-hours versus 152 for 6K-step Std. GRPO, yet achieves CLIP/FID $33.74/15.64$ versus $32.45/17.76$ (Appendix I.2). At matched policy-update count, the measured wall-clock times are 19.0 and 20.5 hours on eight A100-80GB GPUs, with peak memory of 61.95 and 63.82 GB per GPU, respectively. The approximately time-matched checkpoint is selected by proximity to the baseline training budget, not by evaluation scores. Together, these comparisons are consistent with a benefit beyond taking more policy steps, while leaving open comparisons under exact FLOP matching.

Independent evaluation and higher resolution. On 5K paired COCO prompts, ImageReward—unused for training or selection—improves from 0.43 to 0.47 (Appendix I.6). At 512×512, RankE improves CLIP/FID over Std. GRPO from $31.91/33.17$ to $32.46/29.82$ (Appendix I.3).

5 Conclusion

RankE addresses decoder-side Latent Covariate Shift by alternating policy optimization and reward-aware decoder adaptation. The strongest evidence is on LlamaGen-XL: paired runs, fixed-latent interventions, and an approximately time-matched comparison support simultaneous fidelity and alignment gains. Janus-Pro provides more modest evidence of improvement over frozen-decoder RL under matched post-training conditions.

Limitations and future work. The 3.0% peak-memory and 7.9% wall-clock overhead measured at 256×256 may not transfer to higher resolutions. Two paired LlamaGen-XL runs provide limited variance information; Janus-Pro and most additional evaluations are single-run. SFT substantially worsens Janus-Pro FID, and RankE does not recover Base fidelity; corpus mismatch is plausible but not isolated. ImageReward is an independent automated metric, but no human preference study was conducted. The frozen encoder, broader loss-weight and discriminator-schedule sweeps, and evaluation on next-scale architectures remain open. The GEM interpretation is heuristic and gives no convergence guarantee for the practical updates.

Acknowledgments

This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China (NSFC) (Grant No. 62506305), the Zhejiang Leading Innovative and Entrepreneur Team Introduction Program (Grant No. 2024R01007), the Key Research and Development Program of Zhejiang Province (Grant No. 2025C01026), and the Scientific Research Project of Westlake University (Grant No. WU2025WF003).

References

[1] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, 2015. 2

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 2, 4, 15, 23

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. Technical report, OpenAI, 2023. 20, 21

[4] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024. 3

[5] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 1

[6] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *CVPR*, 2022. 15

[7] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. In *ICML*, 2023. 15

[8] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. In *CVPR*, 2025. 15

[9] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 20

[10] 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. 1, 3, 6, 7

[11] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *ICLR*, 2024. 2, 3, 5

[12] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *ICLR*, 2024. 4

[13] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 15

[14] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 1, 6, 15

[15] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In *NeurIPS*, 2023. 3, 7

[16] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *ICML*, 2023. 4, 19

[17] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023. 6, 21

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 5

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 6

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[21] Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *ICML*, 2023. 2, 16

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 19

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017. 2, 4, 15, 16, 23

[24] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025. 1, 3

[25] Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *ICML*, 2018. 2, 16

[26] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 15

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017. 6

[28] Tomasz Korbak, Ethan Perez, and Christopher Buckley. RL with KL penalties is better viewed as Bayesian inference. In *Findings of EMNLP*, 2022. 3, 4, 16

[29] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *ICCV*, 2025. 1, 2, 3, 7, 16

[30] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018. 3, 4, 16

[31] Siyuan Li, Luyuan Zhang, Zedong Wang, Juanxi Tian, Cheng Tan, Zicheng Liu, Chang Yu, Qingsong Xie, Haonan Lu, Haoqian Wang, and Zhen Lei. Mergevq: A unified framework for visual generation and representation with disentangled token merging and quantization. In *CVPR*, 2025. 15

[32] Xinyao Liao, Qiyuan He, Kai Xu, Xiaoye Qu, Yicong Li, Wei Wei, and Angela Yao. Va-π: Variational policy alignment for pixel-aware autoregressive generation. In *CVPR*, 2026. 1, 3, 7

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6, 17, 21

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 7

[35] 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. In *NeurIPS*, 2025. 7

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 19

[37] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 15

[38] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998. 16

[39] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T. Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *CVPR*, 2025. 15

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 7

[41] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. 5

[42] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *ICML*, 2007. 5

[43] 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. In *ICLR*, 2024. 7

[44] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. 2, 3, 5

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6, 20

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 15

[47] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. 2

[48] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019. 1

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 7

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4

[51] 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. 2, 3, 4, 6, 19

[52] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *ICCV*, 2025. 15

[53] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. In *NeurIPS*, 2023. 20, 21

[54] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1, 2, 6, 7

[55] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 6

[56] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. 7, 15, 23

[57] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 1, 15

[58] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, 2024. 3, 7

[59] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025. 1, 3

[60] 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. 7

[61] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992. 4

[62] C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983. 16

[63] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 3, 6, 20, 27

[64] Tianwei Xiong, Jun Hao Liew, Zilong Huang, Jiashi Feng, and Xihui Liu. Gigatok: Scaling visual tokenizers to 3 billion parameters for autoregressive image generation. In *ICCV*, 2025. 15

[65] 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 *NeurIPS*, 2023. 3, 24

[66] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 20, 21

[67] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 7, 15

[68] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024. 15

[69] Guohui Zhang, Hu Yu, Xiaoxiao Ma, Jinghao Zhang, Yaning Pan, Mingde Yao, Jie Xiao, Linjiang Huang, and Feng Zhao. Group critical-token policy optimization for autoregressive image generation. In *ICLR*, 2026. 1, 3, 7

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[71] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. 7

Appendix for RankE

Roadmap

The appendix contains theoretical foundations, diagnostics, and implementation details, followed by the additional controlled evaluations in Appendix I and broader impacts in Appendix J.

Part I – Theoretical Foundations (App. A–B) positions RankE in the literature and develops an idealized GEM-inspired interpretation and states the limits of its connection to the practical algorithm.

Part II – Mechanism Diagnostics and Robustness (App. C–E) measures Latent Covariate Shift, traces training dynamics across reward signals, and consolidates sensitivity studies, answering *whether RankE works for the reasons claimed and is robust to design choices.*

Part III – Reproducibility (App. F–H) records the training algorithm, hyperparameters, compute footprint, and data pipeline, documenting the reported configuration and implementation structure.

Part I | Theoretical Foundations

Where RankE sits in the literature, and why its alternation is principled rather than ad hoc.

A Extended Related Work

This appendix expands §2 along three axes that together motivate RankE's alternating design: discrete visual tokenizers, AR generator factorizations, and the gradient barrier between policy and pixels.

A.1 Discrete Visual Tokenizers

Discrete visual tokenizers compress images into compact grids of latent codes that downstream AR models treat as token sequences. Most adopt vector quantization (VQ) to learn a finite codebook mapping encoder features to discrete indices, as in VQ-VAE [57] and VQGAN [14]. Recent tokenizer research follows three complementary directions: (i) improving the quantization mechanism to reduce VQ-induced fidelity loss [37, 52]; (ii) adding regularization or representation constraints for more structured tokens that better support downstream generation [64, 8]; and (iii) shortening token length via stronger compression for efficiency [31, 68]. All of these works share one assumption with which RankE explicitly breaks: the decoder, once pre-trained, is treated as a fixed renderer.

A.2 AR-based Image Generators

AR-based image generators model images as token sequences under a chosen factorization. Early systems use raster-order decoding with Transformer generators, where scaling yields LLM-like gains [26], exemplified by DALL-E [46], CogView [13], and Parti [67]. Subsequent work explores alternative factorizations: next-scale prediction in VAR [56], masked parallel decoding in MaskGIT [6] and MUSE [7], and randomized orders in RandAR [39]. Despite different decoding schedules, these methods share a two-stage pipeline with a frozen tokenizer: training conditions on ground-truth codes via teacher forcing, while inference conditions on self-sampled codes. As §3 shows, this decoupling becomes a first-order bottleneck *after* preference optimization, motivating our co-trainable decoder.

A.3 The Gradient Barrier in Discrete AR

Direct reward back-propagation does not transfer from diffusion to discrete AR because two non-differentiable operations sit between policy and pixels: categorical sampling [23] and VQ argmax [57]. The straight-through estimator (STE) [2] is a biased surrogate empirically unstable at the codebook

scales ($\sim$16K entries) of modern visual tokenizers [25, 21]; Gumbel-Softmax [23] requires temperature annealing and degrades at high vocabulary sizes. While REPA-E [29] unlocks the VAE for diffusion via reparameterization through a smooth chain, the discrete AR setting requires a categorically different solution: rather than differentiating *through* the discrete bottleneck, RankE alternates *around* it, updating the policy with a discrete-friendly RL objective and the decoder with continuous gradients on policy-sampled latents.

B Idealized GEM-Inspired Interpretation

The following control-as-inference derivation [30, 28] motivates alternating optimization. It is not a convergence proof for the implemented RankE updates.

B.1 Fixed-reference variational objective

Assume $\beta > 0$, a fixed reference policy π_{ref}, and a reward upper bound $r(D_\phi(z), y) \leq C$ that is independent of z, y, ϕ over the considered domain. Define a binary optimality likelihood

$$p(\mathcal{O} = 1 \mid z, y; \phi) = \exp\big((r(D_\phi(z), y) - C)/\beta\big) \leq 1. \tag{9}$$

Writing $Z_\phi(y) = \mathbb{E}_{z\sim\pi_{\text{ref}}}[\exp(r(D_\phi(z), y)/\beta)]$ gives $\log p(\mathcal{O} = 1 \mid y; \phi) = \log Z_\phi(y) - C/\beta$. For a variational policy π_θ with support contained in that of π_{ref}, Jensen's inequality yields

$$\log Z_\phi(y) \geq \mathcal{F}(\theta, \phi; y) := \frac{1}{\beta}\mathbb{E}_{z\sim\pi_\theta}[r(D_\phi(z), y)] - D_{\text{KL}}(\pi_\theta(\cdot \mid y)\|\pi_{\text{ref}}(\cdot \mid y)). \tag{10}$$

The gap is $D_{\text{KL}}(\pi_\theta\|p^*_\phi)$, where $p^*_\phi(z \mid y) = \pi_{\text{ref}}(z \mid y)\exp(r(D_\phi(z), y)/\beta)/Z_\phi(y)$. Thus Eq. 10 is an exact variational identity under these assumptions. For a fixed decoder penalty $\Omega(\phi)$, the idealized regularized objective is

$$\mathcal{J}_{\text{ideal}}(\theta, \phi) = \mathbb{E}_y[\mathcal{F}(\theta, \phi; y)] - \Omega(\phi). \tag{11}$$

If $\exp(-\Omega)$ defines a proper prior, this has a MAP interpretation; otherwise it remains a regularized variational objective. With the reference and penalty held fixed, block updates that increase this same objective give a GEM-inspired coordinate-ascent interpretation [38, 62]. Such improvement is a condition, not an automatic property of arbitrary gradient steps.

B.2 Why the practical algorithm is approximate

The implemented policy objective (Eq. 3) uses group-normalized finite-sample advantages and PPO clipping. A sample's inclusion in its own group baseline, division by a random group standard deviation, and clipping generally change the estimator; they do not give an unbiased gradient of Eq. 11 or a guaranteed trust-region projection. Updating the EMA policy also changes the reference objective over time.

For differentiable rewards, direct reward backpropagation follows the reward term's decoder gradient with detached latents. Rank-GAN instead uses reward-conditioned adversarial realism gradients. Its weights are not exact importance weights for p^*_ϕ: exact weights would include $\pi_{\text{ref}}(z \mid y)/\pi_\theta(z \mid y)$ as well as the exponential reward factor. When $\pi_\theta = p^*_\phi$, exact importance weights would be constant, not exponential in reward. The evolving discriminator, EMA teacher, reconstruction GAN term, and separate loss weights further distinguish the practical objective from a fixed MAP ELBO.

Consequently, RankE admits an *approximate GEM-inspired block-coordinate interpretation*. We do not claim monotonic likelihood/ELBO improvement or convergence to a stationary point for the practical stochastic, clipped, adversarial updates. The empirical curves describe observed training behavior rather than establish those guarantees.

Part II | Mechanism Diagnostics and Robustness

Empirical evidence that RankE's gains come from the mechanism it was designed to engage, and that the design is robust within reasonable ranges.

C Latent Covariate Shift: Measurement Protocol

This appendix details how the Latent Covariate Shift diagnostic reported in Fig. 1 and Fig. 6 is computed, so that the curves are reproducible from token statistics alone.

Token extraction protocol. *Ground-truth tokens.* We sample N=5,000 images from MS-COCO 2014 validation [33] and encode via the frozen VQ tokenizer: $\mathbf{z} = \arg\min_{c \in \mathcal{C}} \|\mathrm{Enc}(x)_{ij} - c\|_2$. *Policy-sampled tokens.* For each checkpoint $\pi_\theta^{(t)}$ at $t \in \{500, 1000, \ldots, 6000\}$, we generate N=5,000 sequences from the same COCO prompts using nucleus sampling (p=0.9, temperature 1.0, top-k=1000, CFG 6.0, fixed seed). We record raw token indices without pixel decoding to isolate the latent distribution from decoder artifacts.

Global marginal codebook frequencies. For $N = 5{,}000$ generated sequences of length L and codebook entry c, define

$$p_\theta(c) = \frac{1}{NL} \sum_{n=1}^{N} \sum_{\ell=1}^{L} \mathbf{1}[z_{n\ell} = c], \qquad D_{\mathrm{KL}}(p_\theta \| p_{\mathrm{GT}}) = \sum_c p_\theta(c) \log \frac{p_\theta(c)}{p_{\mathrm{GT}}(c)}. \tag{12}$$

The ground-truth frequencies p_{GT} are computed analogously using the same frozen tokenizer. The diagnostic is an aggregate of codebook usage over positions and prompts; it is not a sequence-level policy KL, a position-wise KL, or a prompt-conditional estimate. Its aggregation reduces sampling noise but discards co-occurrence and spatial information. The marginal-KL implementation uses natural logarithms and the stabilized statistic $\sum_c p_\theta(c) \log[(p_\theta(c)+\varepsilon)/(p_{\mathrm{GT}}(c)+\varepsilon)]$ with $\varepsilon = 10^{-10}$, adding ε inside the ratio without renormalizing the frequencies. This avoids infinite terms from unobserved reference codes. Token entropy is computed in bits using $\log_2$.

Position-wise Jensen–Shannon divergence. At each token position ℓ, let $p_{\theta,\ell}$ and $p_{\mathrm{GT},\ell}$ be empirical code frequencies across samples. We average

$$\frac{1}{L} \sum_{\ell=1}^{L} \mathrm{JS}(p_{\theta,\ell}, p_{\mathrm{GT},\ell}), \quad \mathrm{JS}(p, q) = \tfrac{1}{2} D_{\mathrm{KL}}(p \| m) + \tfrac{1}{2} D_{\mathrm{KL}}(q \| m), \quad m = \tfrac{1}{2}(p + q). \tag{13}$$

This captures position-conditioned marginal shift, while still omitting higher-order co-occurrences and prompt conditioning. Appendix I.4 complements these diagnostics with decoder and token interventions.

D Training Dynamics and Convergence Behavior

Having defined the diagnostic, we now ask how the loss components themselves behave during co-evolution. This appendix reports per-step dynamics under both reward signals (§D.1) and convergence patterns across co-evolution rounds (§D.2).

D.1 Per-Step Dynamics under CLIP and HPSv2

Fig. D.1 tracks six diagnostics over 3,000 steps on LlamaGen-XL with identical hyperparameters under CLIP and HPSv2 rewards. Across all panels, the two reward settings show qualitatively similar convergence patterns:

- **Decoder fidelity (a, b).** Reconstruction loss on policy-sampled images stays near zero (HPSv2) or briefly spikes around steps 1,000–2,000 before recovering (CLIP); GT reconstruction stays in 0.3–0.4, with HPSv2 marginally higher—a mild trade-off, not destabilization.
- **Adversarial dynamics (c, d).** The discriminator reaches $\sim$0.5 within 500 steps (declining to $\sim$0.35 under CLIP), while consistency loss settles near 0.01 after early transients, consistent with EMA distillation limiting abrupt decoder changes.
- **Policy-side dynamics (e, f).** KL rises to 2.5–3.0 under both rewards with nearly identical trajectories, and rewards increase over training (CLIP $\sim$21$\rightarrow$29, HPSv2 $\sim$25.5$\rightarrow$27.5).

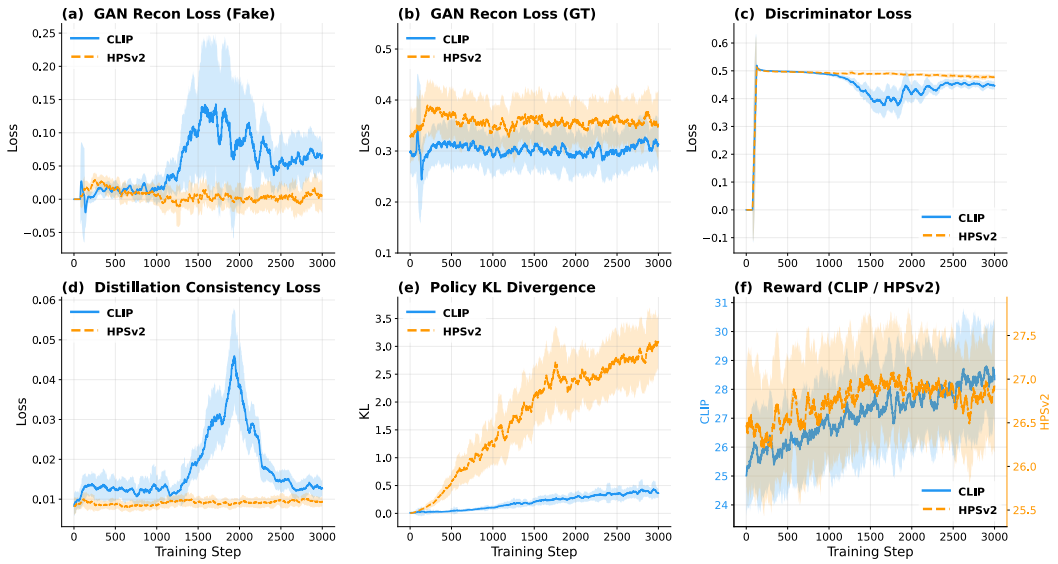

Figure D.1: Training dynamics of RankE over 3,000 steps under **CLIP** (blue, solid) and **HPSv2** (orange, dashed). (a, b) GAN reconstruction loss on generated and ground-truth images. (c, d) Discriminator and distillation consistency losses. (e) Policy KL divergence. (f) Reward curves with dual y-axes.

D.2 Convergence across Co-Evolution Rounds

The reported curves show rising rewards and evolving reconstruction, adversarial, and consistency losses (Fig. D.1), alongside an overall FID improvement for RankE over the checkpoints in Fig. 4(b). Fig. D.1(b) measures ground-truth reconstruction loss, not FID. These curves do not separately resolve FID changes within policy and decoder stages or establish convergence of the alternating updates.

E Extended Ablations and Sensitivity Studies

The main ablations isolate *which* components matter; this appendix probes *how sensitive* each component is to its hyperparameter. We report exploratory sweeps of consistency weight, reward-weighting temperature, and EMA decay on LlamaGen-XL with CLIP reward (MS-COCO 30K). These do not constitute a joint search over discriminator schedules and all loss weights. The reported sweeps cover a limited range of settings and do not establish robustness outside that range.

E.1 Consistency Distillation Weight λ_c

We ablate λ_c across λ_c=10 (Run A), λ_c=1 (Run B, default), and λ_c=50 (Run C). Dynamics are in Fig. E.1; the recorded checkpoint metrics are in Table E.1. These exploratory runs are separate from the main full-RankE result and should not be read as its matched final-checkpoint ablation. Run B yields the most stable training. Run A introduces mild drift but achieves the best FID/GenEval. Run C causes decoder collapse around step 1,500 (GT reconstruction diverges, discriminator drops to near zero), showing that excessive consistency overwhelms adversarial signal. We adopt λ_c=1 for overall stability.

Table E.1: Ablation on λ_c

Setting	λ_c	**CLIP** ↑	**FID** ↓	**GenEval** ↑
Run A	10	33.64	**17.37**	**0.425**
Run B (default)	1	**34.13**	18.78	0.394
Run C	50	32.98	17.75	0.416

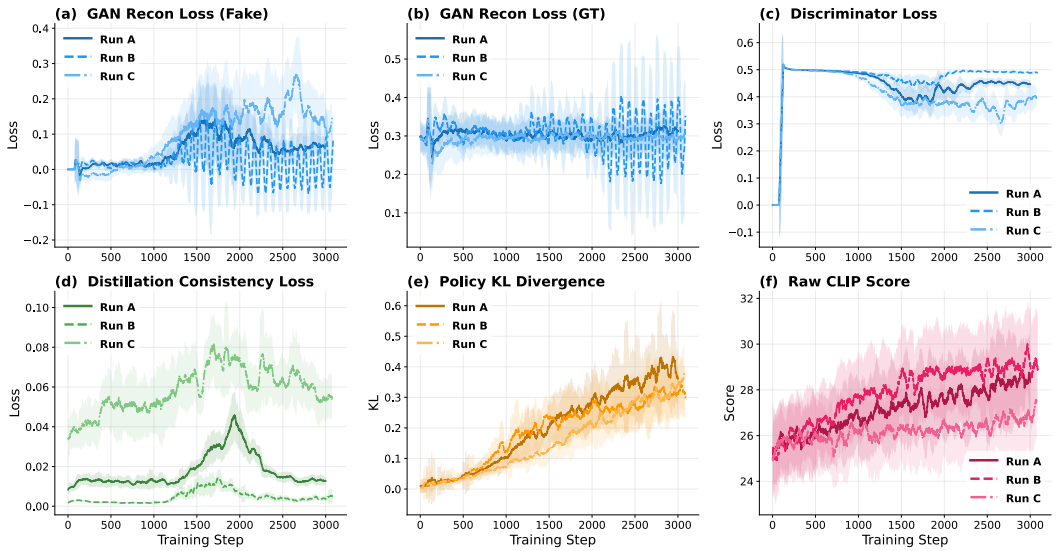

Figure E.1: Training dynamics for the λ_c ablation. Run A: λ_c=10; Run B: λ_c=1 (default); Run C: λ_c=50. Excessive λ_c causes decoder collapse (Run C); moderate values stay stable.

E.2 Reward Weighting and EMA Sensitivity

Table E.2 summarizes the temperature and EMA sweeps across CLIP, FID, and GenEval. Of the tested settings, $\tau = 0.1$ and $\alpha = 0.999$ give the best values on all three metrics. A smaller temperature (0.01) concentrates reward weights and gives an FID of 16.12; a larger value (1.0) softens weighting but is not exactly uniform, which requires $\tau \to \infty$ or equal rewards. The faster EMA teacher ($\alpha = 0.900$) gives weaker results than the slower references in this sweep. These limited sweeps do not establish robustness to arbitrary temperatures, decay rates, or discriminator schedules.

Table E.2: Reward-weighting and EMA sensitivity on LlamaGen-XL. Bold settings are the defaults. Each entry is a single-run measurement; no run-to-run uncertainty is available.

Sweep	Setting	CLIP ↑	FID ↓	GenEval ↑
Temperature	$\tau = 0.01$	33.49	16.12	0.408
Temperature	$\tau = 0.1$ **(default)**	**33.76**	**15.21**	**0.425**
Temperature	$\tau = 1.0$	33.43	15.65	0.410
EMA decay	$\alpha = 0.900$	33.48	15.75	0.412
EMA decay	$\alpha = 0.990$	33.63	15.45	0.419
EMA decay	$\alpha = 0.999$ **(default)**	**33.76**	**15.21**	**0.425**

Part III | Reproducibility

Reported implementation details: hyperparameters, pseudocode, and data pipeline.

F Implementation Details and Hyperparameters

Implementation. We implement RankE in PyTorch and train on 8×NVIDIA A100 GPUs (bf16). Stage 1 optimizes the AR policy with GRPO [51] at learning rate 1×10^{-5}, with the KL coefficient β scheduled against an EMA reference policy to suppress reward exploitation [16]. Stage 2 freezes the policy and updates the decoder with AdamW [36] at 5×10^{-5}, supervised by a PatchGAN discriminator [22] and an EMA teacher with decay 0.999.

Training hyperparameters. Table F.1 records the reported main-paper defaults for both stages.

Table F.1: Hyperparameters for Stage 1 (Policy Alignment) and Stage 2 (Decoder Adaptation).

Stage 1: Policy Alignment			
Hyperparameter	**Value**	**Hyperparameter**	**Value**
Optimizer	AdamW	Learning rate	1×10^{-5}
Weight decay	0.01	GRPO group size G	8
PPO clip ϵ	0.2	KL penalty β	0.05
Sampling temp.	1.0	Top-p	0.9
Mixed precision	`bf16`		
Stage 2: Decoder Adaptation			
Hyperparameter	**Value**	**Hyperparameter**	**Value**
Optimizer	AdamW	Learning rate	5×10^{-5}
Weight decay	0.05	(β_1, β_2)	$(0.5, 0.9)$
Discriminator	PatchGAN	Discriminator lr	5×10^{-5}
EMA decay α	0.999	Weight temperature τ	0.1
λ_{r}	1.0	λ_{g}	0.5
λ_{c}	1.0	λ_{d}	0.1
Mixed precision	`bf16`		

VQ tokenizer architecture. LlamaGen-XL and Janus-Pro share a VQ-VAE tokenizer with $16\times$ spatial downsampling. The codebook has $|\mathcal{C}|$=16,384 entries, embedding dimension d=8, ℓ_2-normalized before lookup. The encoder/decoder is a convolutional ResNet with channel multipliers $[1, 1, 2, 2, 4]$ and self-attention at 16×16. A 256×256 input yields a 256-token raster-scan sequence.

Reward models. We use two reward models: *CLIP Score* [45] (ViT-g/14 cosine similarity, scaled by 100) and *HPSv2* [63] (a human-preference scorer trained on large-scale pairwise judgments). The reward models are frozen. Under CLIP, reward gradients flow through the decoder but not through detached token indices. HPSv2 is used as scalar feedback in the black-box setting; no direct scorer gradient is required by Rank-GAN.

Compute footprint. The measured 256×256 comparison uses eight NVIDIA A100-SXM4-80GB GPUs, bf16, global batch size 32, group size 8, rollout length 256, and 6,000 policy-training steps. Std. GRPO uses 61.95 GB peak memory per GPU and 19.0 h; RankE uses 63.82 GB and 20.5 h. The increases are 1.87 GB (3.0%) and 1.5 h (7.9%). Appendix I.2 reports throughput and the separate approximate time-budget comparison. These measurements do not establish higher-resolution overhead or total project compute.

G Training Algorithm

Algorithm 1 summarizes the end-to-end training procedure. The framework alternates between policy optimization for the AR generator and decoder refinement under the multi-objective loss in Eq. 4.

H Training Data and Curation

Training data. We construct a 15K training set from the publicly available BLIP3o-60k instruction-tuning dataset [9] (high-quality image–text pairs curated with GPT-4o across diverse scenes, objects, and human gestures). Two-stage curation adapts this pool for reward-based post-training. (1) Long captions from synthetic sources (DALL-E 3 [3], JourneyDB [53]) often exceed the 77-token CLIP context window, causing reward truncation and unstable gradients; we use Qwen2.5-Instruct [66] to compress each caption into concise visual tags ($\leq$ 50 words), retaining key attributes (subject, style, lighting, color) while removing prose. (2) Stratified sampling balances domain coverage and evaluation-relevant capabilities (Table H.1).

Algorithm 1 RankE: alternating policy and decoder adaptation

Require: Paired data (y, x_{gt}); frozen encoder E and quantizer Q; policy π_θ; decoder D_ϕ; discriminator Disc_ψ; frozen reward r
Require: Rounds K, stage lengths n_π, n_D, group size G, clip ϵ, KL weight β, temperature τ, decoder weights $\lambda_r, \lambda_g, \lambda_c, \lambda_d$
Initialize EMA reference parameters $\theta_{\text{ref}} \leftarrow \theta$, $\phi_{\text{ema}} \leftarrow \phi$
Initialize separate AdamW optimizers for θ, ϕ, and ψ
for $k = 1, \ldots, K$ **do**
  Stage 1: freeze decoder and discriminator; update policy
  for $t = 1, \ldots, n_\pi$ **do**
    Sample prompts y; snapshot rollout policy $\theta_{\text{old}} \leftarrow \text{sg}[\theta]$
    Draw $\{z_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid y)$; $r_i \leftarrow r(D_\phi(z_i), y)$
    Compute detached $A_i = (r_i - \mu_r)/\sigma_r$ if $\sigma_r > 0$; otherwise set $A_i = 0$
    Form importance ratios $\rho_i = \pi_\theta(z_i \mid y)/\pi_{\theta_{\text{old}}}(z_i \mid y)$
    Compute clipped GRPO and EMA-reference KL using Eq. 3
    Update only θ with its AdamW optimizer; update θ_{ref} by EMA
  end for
  Freeze the end-of-stage policy snapshot $\bar{\theta} \leftarrow \text{sg}[\theta]$
  Stage 2: freeze policy; update decoder and discriminator
  for $t = 1, \ldots, n_D$ **do**
    Sample (y, x_{gt}) and fresh detached $\{\hat{z}_i\}_{i=1}^{G} \sim \pi_{\bar{\theta}}(\cdot \mid y)$
    $z_{\text{gt}} \leftarrow \text{sg}[Q(E(x_{\text{gt}}))]$; $\hat{x}_i \leftarrow D_\phi(\hat{z}_i)$
    $r_i \leftarrow r(\hat{x}_i, y)$; $w_i \leftarrow \text{sg}[G \exp(r_i/\tau)/\sum_j \exp(r_j/\tau)]$
    Update only ψ with its adversarial discriminator loss on real and detached fake images
    Freeze ψ for the decoder update (retain image-input gradients)
    $\mathcal{L}_{\text{Rank-GAN}} \leftarrow -G^{-1} \sum_i w_i \text{Disc}_\psi(\hat{x}_i)$
    Compute GT reconstruction including its GAN term (Eq. 7)
    Compute LPIPS consistency to $\text{sg}[D_{\phi_{\text{ema}}}(\hat{z}_i)]$ (Eq. 8)
    Compute direct reward loss (Eq. 5) when enabled; otherwise $\lambda_d = 0$
    Combine all four decoder terms using Eq. 4
    Update only ϕ with its AdamW optimizer; update ϕ_{ema} by EMA
  end for
end for
return π_θ, D_ϕ

Table H.1: Composition of the 15K training set.

Source	Domain	Count	Ratio
JourneyDB [53]	Aesthetics / artistic styles	4,500	30%
DALL-E 3 [3]	Complex attribute binding	3,000	20%
MS-COCO [33]	Natural scenes	3,000	20%
Human Gestures	Structural accuracy	3,000	20%
GenEval [17]	Compositional skills	750	5%
Typography	Text rendering	750	5%
Total		**15,000**	**100%**

Caption summarization pipeline. We apply Qwen2.5-Instruct (7B) [66] with the prompt:

```
Summarize the following image caption into concise visual tags (≤50
words). Preserve: subjects, styles, lighting, colors, composition.
Remove: narrative prose, subjective descriptions, filler words.

Caption: {original_caption}

Visual Tags:
```

This compression preserves semantic content relevant for reward evaluation while keeping the full caption within CLIP's effective context window.

I Additional Controlled Evaluations

Unless stated otherwise, additional results are single-run estimates. Main tables retain the original representative run; only Appendix I.1 aggregates training runs.

I.1 Paired-run reproducibility on LlamaGen-XL

We add one independent paired training seed, giving two pairs in total. Within each pair, Std. GRPO and RankE share the SFT checkpoint, initialization seed, data order, sampling setup, preprocessing, and FID reference statistics. We report mean $\bar{x}$ and sample standard deviation $s = \sqrt{\sum_{i=1}^{2}(x_i - \bar{x})^2/(2-1)}$. Differences are formed within each pair before aggregation. These deviations describe variation across two training/evaluation runs, not standard errors or confidence intervals. Both pairs improve both metrics. Two runs provide a reproducibility check, not a significance test or a precise variance estimate. Janus-Pro is single-run and is not covered by these error bars.

Table I.1: Two paired LlamaGen-XL runs (mean $\pm$ sample SD; differences computed within pairs).

Method	CLIP ↑	FID ↓
Std. GRPO	32.41 ± 0.06	17.85 ± 0.12
RankE	33.67 ± 0.13	15.37 ± 0.22
Paired difference	$+1.26 \pm 0.07$	-2.48 ± 0.10

Table I.2: Measured cost at 6,000 policy updates (256×256, eight A100 GPUs). Hours denote wall-clock time; GPU-h sums time over GPUs; Steps/h counts policy updates including decoder work.

Method	Steps	GB/GPU	Hours	GPU-h	Steps/h
Std. GRPO	6,000	61.95	19.0	152	315.8
RankE	6,000	63.82	20.5	164	292.7

I.2 Measured cost and approximately matched training time

The measured comparison uses eight NVIDIA A100-SXM4-80GB GPUs, bf16 precision, global batch size 32, group size 8, rollout length 256, and 6,000 policy-training steps. GPU-hours equal elapsed hours times eight; throughput counts policy-training steps per elapsed hour, including decoder work for RankE. At equal policy-update count, RankE adds 1.87 GB peak memory per GPU (3.0%) and 1.5 h (7.9%), leaving nominal headroom of 16.18 GB relative to 80 GB. Autoregressive rollout and policy optimization dominate cost; alternation avoids retaining policy and decoder backward graphs simultaneously.

For the separate comparison in Table I.3, the saved RankE checkpoint is selected solely as the nearest to the measured 19-hour Std. GRPO budget, not by evaluation scores. Measured end-to-end throughput gives $20.5 \times 5500/6000 \approx 18.79$ hours and 150.3 GPU-hours for 5,500 steps. The two operating points are CLIP/FID $32.45/17.76$ for Std. GRPO and $33.74/15.64$ for RankE. RankE uses approximately 1.1% fewer GPU-hours, improves CLIP by 1.29, and reduces FID by 2.12. This approximate wall-clock/GPU-hour match complements the equal-step comparison; neither is strictly FLOP-matched.

I.3 Janus-Pro initialization and higher resolution

Janus-Pro operates at 384×384. Its Base-to-SFT FID change ($18.95 \rightarrow 26.73$) precedes RL. Corpus mismatch is a plausible contributor, but this does not isolate which SFT data properties cause the change. Starting from the same SFT checkpoint, RankE improves CLIP/FID from $33.60/25.59$ (Std. GRPO) to $33.86/25.19$, without recovering Base FID. Table I.3 gives the separate Base-initialized comparison, which removes the preceding SFT stage. RankE again improves over frozen-decoder RL and obtains FID within 0.05 of Base in this run. This small difference does not establish statistical equivalence to Base. The cross-backbone conclusion is improvement over frozen-decoder RL under matched post-training conditions.

Table I.3 also reports the 512×512 comparison. At this resolution, LlamaGen-XL's $16\times$ downsampling yields 1,024 tokens, four times the sequence length at 256×256. RankE gains 0.55 CLIP and reduces FID by 3.35 (10.1%) relative to Std. GRPO. Here Std. GRPO already improves FID over Base: this is evidence of an additional co-evolution benefit, not a universal claim that policy-only RL worsens fidelity.

Table I.3: Cross-setting comparisons from the controlled evaluation. The time-matched RankE checkpoint is the saved checkpoint nearest the 19-hour Std. GRPO budget, selected without using evaluation scores; its elapsed time and GPU-hours are estimated from measured throughput. All entries are single-run estimates.

(a) Backbone and resolution comparisons.

Setting	Method	CLIP ↑	FID ↓
Janus-Pro 384^2, no SFT	Base	33.20	18.95
	Std. GRPO	33.50	19.30
	RankE	33.65	19.00
LlamaGen-XL 512^2	Base	30.62	46.29
	Std. GRPO	31.91	33.17
	RankE	32.46	29.82

(b) Approximately time-matched LlamaGen-XL comparison at 256^2.

Method	Policy updates	Wall-clock h	CLIP ↑	FID ↓
Std. GRPO	6,000	19.0	32.45	17.76
RankE	5,500	≈ 18.8	33.74	15.64

Alternative tokenizers and gradient surrogates. RankE requires computable policy rollout log-probabilities and a differentiable decoder over completed discrete representations. A next-scale policy such as VAR [56] could sum log-probabilities over a multi-scale token pyramid and adapt the decoder on detached completed pyramids. This is structural compatibility, not empirical validation: no VAR results are reported. Longer sequences and larger decoder activations may increase cost and affect stability; the measured 256×256 overhead is not assumed to transfer. STE and Gumbel-Softmax [2, 23] offer alternative approximate gradients through discrete sampling. Table 4 isolates the policy and decoder roles, but is not a compute-matched comparison against those methods, over which we do not claim universal dominance.

I.4 Mechanistic diagnostics and interventions

Structured shift and the decoder contribution. Table I.4 combines two complementary tests. The position-wise diagnostic from Appendix C increases after RL. RankE retains almost the same JS as Std. GRPO (0.0459 vs. 0.0468), yet reduces FID from 17.76 to 15.21: improved fidelity need not require restoring the latent distribution. The fixed-latent intervention then holds prompts, policy outputs, token sequences, sampling seeds, and evaluation settings fixed, changing only the decoder. CLIP improves from 33.32 to 33.71 and FID from 17.63 to 15.28, isolating the rendering contribution. These intervention scores are distinct from the main full-system comparison.

Table I.4: Position-wise latent shift and fixed-latent decoder intervention. The decoder intervention changes only the decoder while holding prompts, policy outputs, token sequences, sampling seeds, and evaluation settings fixed. All entries are single-run estimates.

Analysis	Setting	Position-wise JS	CLIP ↑	FID ↓
Structured shift	SFT	0.0314	–	16.58
	Std. GRPO	0.0468	–	17.76
	RankE	0.0459	–	15.21
Fixed-latent decoder swap	Frozen decoder	–	33.32	17.63
	Adapted decoder	–	33.71	15.28

Controlled nearest-neighbor perturbation. Starting with the same SFT latents, randomly selected token positions are replaced by codes sampled uniformly from their top-20 nearest neighbors in the frozen VQ codebook embedding space, excluding the original code. Prompts, original latents, replacement seeds, decoder parameters, and evaluation settings are controlled. Table I.5 shows that

global KL and FID rise monotonically with replacement ratio while CLIP decreases by only 0.11. This shows that local decoder-input perturbations can degrade pixel fidelity without comparably changing prompt alignment. The intervention does not exactly reproduce the RL distribution, and KL alone does not identify the causal contribution of every changed token property.

I.5 Separating Rank-GAN from reward backpropagation

Table I.6 presents a step- and schedule-matched 2×2 ablation, retaining reconstruction and consistency. Uniform GAN assigns equal weights; Rank-GAN uses detached normalized reward weights (Eq. 6). Without reward backpropagation, uniform GAN and Rank-GAN obtain CLIP/FID $31.95/20.84$ and $32.53/20.16$; with it enabled, they obtain $33.48/15.92$ and $33.76/15.21$. Direct reward backpropagation supplies the dominant semantic-alignment gain. With it enabled, reward weighting additionally improves CLIP by 0.28 and FID by 0.71. Rank-GAN is reward-conditioned realism adaptation, with an indirect alignment benefit from rendering; it supplies no separate semantic target. This matches steps and schedule, not measured FLOPs. The leave-one-out ablations address component necessity under a common configuration, not comprehensive hyperparameter sensitivity.

Table I.5: Top-20 nearest-neighbor token perturbation (single run, fixed SFT latents and decoder). Global KL aggregates code frequencies.

Replaced tokens (%)	Global KL	CLIP ↑	FID ↓
0	0.0985	31.86	16.58
5	0.1047	31.83	16.92
20	0.1186	31.75	17.99

Table I.6: Policy-update- and schedule-matched 2×2 decoder-objective ablation (single run; reconstruction and consistency retained).

GAN objective	Reward backpropagation	CLIP ↑	FID ↓
Uniform	Off	31.95	20.84
Rank-GAN	Off	32.53	20.16
Uniform	On	33.48	15.92
Rank-GAN	On	33.76	15.21

I.6 Independent reward evaluation

We evaluate CLIP-trained checkpoints with ImageReward [65], which is not used for training, checkpoint selection, or hyperparameter tuning. Both methods use the same 5,000 MS-COCO prompts, sampling parameters, and paired sampling seeds. ImageReward improves from 0.43 (Std. GRPO) to 0.47 (RankE), a gain of 0.04. This transfers the gain beyond the optimized scorer, but remains an automated single-run evaluation without a reported uncertainty estimate. It does not rule out all reward-specific artifacts; human evaluation remains future work.

J Broader Impacts

Better fidelity and prompt adherence may help creative tools and visual communication, but may also facilitate deceptive synthetic media, impersonation, and amplification of data or reward-model biases. Post-training can alter inherited safeguards. This work does not measure safety or establish deployment safeguards; human evaluation, bias assessment, and appropriate usage policies remain necessary considerations before deployment.

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 Introduction scope the strongest evidence to LlamaGen-XL and the Janus-Pro result to improvement over frozen-decoder RL under matched post-training conditions.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5 discusses compute, limited paired runs, Janus-Pro initialization, automated evaluation, tokenizer coverage, and the approximate theoretical interpretation.

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 B derives the idealized variational bound under explicit fixed-reference and bounded-reward assumptions, and distinguishes it from practical updates for which no convergence guarantee is claimed.

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: [No]

   Justification: The paper supplies objectives, algorithm structure, model and data descriptions, and reported hyperparameters. Exact run identifiers, full alternating-stage lengths, and all evaluation implementation conventions are not documented sufficiently to claim complete reproducibility.

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: The project page is linked on the title page. This manuscript does not establish that a complete public package reproducing every main and additional experiment is available. Data curation is described in Appendix H.

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: [No]

   Justification: Sections 4 and Appendix F document reported settings, but the complete discriminator schedule, stage lengths, seeds, and evaluation implementation details are not all specified.

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: [Yes]

Justification: Appendix I.1 reports means and sample standard deviations over two paired LlamaGen-XL runs, including paired differences and the variability represented. These are a reproducibility check, not significance tests; Janus-Pro and other additional results are single-run estimates.

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: [No]

   Justification: Appendix I.2 gives hardware, precision, per-GPU peak memory, measured time, throughput, and GPU-hours for the main comparison. Per-experiment costs for all additional runs and total project compute, including exploratory runs, are not available.

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 acknowledge the NeurIPS Code of Ethics and academic integrity obligations. No human-subject study was conducted; limitations of evaluation and potential misuse are discussed in Section 5 and Appendix J.

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 J discusses potential creative benefits, deceptive media, bias, and limitations of inherited safeguards.

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: [No]

    Justification: The method adapts image generators and can affect their behavior. No new deployment safeguards or post-training safety evaluation are established in this work; Appendix J discusses this limitation.

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: [No]

    Justification: Original models and datasets are cited, but this manuscript does not enumerate and verify the exact license and terms of every reused asset and source image. Public availability alone does not establish an open license.

13. **New assets**

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

    Answer: [No]

Justification: The method introduces training code and adapted model parameters, but a complete released package with all checkpoint, data-provenance, and licensing documentation is not established here.

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: This work does not involve crowdsourcing or research with human subjects. HPSv2 reward is a pre-trained scoring model from prior work [63], not collected by the authors.

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: This work does not involve crowdsourcing or research with human subjects.

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: Appendix H describes Qwen2.5-Instruct caption summarization, including the prompt. LLM assistance was also used for editing, understanding technical concepts, and implementing standard methods; it does not substitute for author verification of claims or references.