VA-π: Variational Policy Alignment for Pixel-Aware Autoregressive Generation

Supplementary Material

A. Theoretical Details of VA-π

In this section, we (1) restate the proof of the ELBO with respect to the random variable and posterior defined in Sec. A.1; (2) provide additional insights into its relationship with the VAE [10] and autoregressive (AR) models based on VQVAE [18] in Sec. A.2; and (3) further justify the formulation of the prior regularization term in Sec. A.3.

A.1. Proof of Alignment ELBO

While the proof of ELBO is already provided in existing literature, such as VAE [10]. We restate it with posterior based on a discrete token sequence as latent variable.

Setup. Let $\mathbf{I}$ denote the observed image and $\mathbf{x} \in \mathcal{X}$ the discrete token sequence generated from an autoregressive model. We assume a generative model with parameters θ as:

$$p_\theta(\mathbf{I}, \mathbf{x}) \;=\; p_\theta(\mathbf{I} \mid \mathbf{x})\, p(\mathbf{x}), \tag{12}$$

where $p(\mathbf{x})$ is a prior over token sequences (e.g., the AR prior) and $p_\theta(\mathbf{I} \mid \mathbf{x})$ is a pixel-space likelihood (e.g., the tokenizer decoder composed with a reconstruction distribution). Our training objective is the marginal log-likelihood:

$$\log p_\theta(\mathbf{I}) \;=\; \log \sum_{\mathbf{x}\in\mathcal{X}} p_\theta(\mathbf{I}, \mathbf{x}) \quad \text{(integral for continuous } \mathbf{x}\text{)}, \tag{13}$$

which is generally intractable to evaluate directly because summing/integrating over $\mathbf{x}$ is prohibitive.

Introducing a variational posterior. Let $q_\phi(\mathbf{x} \mid \mathbf{I})$ be any distribution supported on $\mathcal{X}$ (in practice, produced by teacher forcing so that it is easy to sample from and to evaluate). Multiply and divide the integrand in (13) by $q_\phi(\mathbf{x} \mid \mathbf{I})$:

$$\log p_\theta(\mathbf{I}) = \log \sum_{\mathbf{x}} q_\phi(\mathbf{x} \mid \mathbf{I}) \frac{p_\theta(\mathbf{I}, \mathbf{x})}{q_\phi(\mathbf{x} \mid \mathbf{I})}. \tag{14}$$

Jensen's inequality. We now apply Jensen's inequality to the concave function $\log(\cdot)$:

$$\log \mathbb{E}_{q_\phi}\big[f(\mathbf{x})\big] \;\geq\; \mathbb{E}_{q_\phi}\big[\log f(\mathbf{x})\big] \qquad (\text{for } f(\mathbf{x}) > 0).$$

Using $f(\mathbf{x}) = \frac{p_\theta(\mathbf{I},\mathbf{x})}{q_\phi(\mathbf{x}|\mathbf{I})}$ in (14) gives:

$$\log p_\theta(\mathbf{I}) \geq \sum_{\mathbf{x}} q_\phi(\mathbf{x} \mid \mathbf{I}) \log \frac{p_\theta(\mathbf{I}, \mathbf{x})}{q_\phi(\mathbf{x} \mid \mathbf{I})} \tag{15}$$

$$= \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log p_\theta(\mathbf{I}, \mathbf{x})\big] - \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log q_\phi(\mathbf{x} \mid \mathbf{I})\big] \tag{16}$$

$$= \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log p_\theta(\mathbf{I} \mid \mathbf{x})\big] + \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log p(\mathbf{x})\big] \tag{17}$$

$$- \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log q_\phi(\mathbf{x} \mid \mathbf{I})\big] \tag{18}$$

$$= \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log p_\theta(\mathbf{I} \mid \mathbf{x})\big] - \mathrm{KL}(q_\phi(\mathbf{x} \mid \mathbf{I}) \,\|\, p(\mathbf{x}))\,. \tag{19}$$

We define the right-hand side of (19) as the *evidence lower bound (ELBO)*:

$$\mathcal{L}(\theta, \phi; \mathbf{I}) \;\triangleq\; \mathbb{E}_{q_\phi(\mathbf{x}|\mathbf{I})}\big[\log p_\theta(\mathbf{I} \mid \mathbf{x})\big] - \mathrm{KL}(q_\phi(\mathbf{x} \mid \mathbf{I}) \,\|\, p(\mathbf{x}))\,, \tag{20}$$

so that $\log p_\theta(\mathbf{I}) \geq \mathcal{L}(\theta, \phi; \mathbf{I})$.

Optimization. When $q_\phi(\mathbf{x} \mid \mathbf{I})$ perfectly matches the true posterior $p_\theta(\mathbf{x} \mid \mathbf{I})$, the KL term becomes zero, and maximizing the ELBO is equivalent to maximum likelihood estimation (MLE) as proved in existing literature [10, 18, 61].

A.2. Comparison with VAE and AR on VQVAE

Our theoretical framework can be justified from the perspective of VAE and AR based on VQVAE. Notice that variational optimization is independent of the choice of the latent variable, posterior and prior. Specifically, the ELBO of **VA-π** can be seem as a variant of VQVAE's ELBO with redefined prior and posterior as shown in Tab. 6.

VAE [10] treats the latent variable is continuous feature, with prior in standard Gaussian distribution and the posterior is parameterized by the encoder. During sampling, the model firstly sample noise from standard Gaussian distribution and forward it to decoder to get the generated image.

VQVAE used in standard visual AR generation [12, 18] instead treats the latent variable as a discrete token sequence, consistent with our formulation. The key distinction, however, lies in how the posterior and prior are defined. In VQVAE, the posterior is determined solely by the encoder and quantizer: given a reference image, the model produces a deterministic token sequence that can be viewed as sampling from a delta distribution, i.e., $q_\phi(\mathbf{x} \mid \mathbf{I})$ is one-hot. The prior, in contrast, is assumed to follow a uniform categorical distribution. Under these assumptions, the KL divergence term becomes a constant independent of learnable parameters and is thus omitted from the objective. However, since the mismatch exists between the assumed uni-

Formulation	$q(\mathbf{x} \mid \mathbf{I})$ (Posterior)	$p(\mathbf{x})$ (Prior)	Trainable Modules	Objective
VAE	Continuous Gaussian Distribution	Gaussian $\mathcal{N}(0, I)$	Encoder, Decoder	Reconstruction + Generation
VQVAE	Categorical Distribution (Dirac)	Uniform categorical	Encoder, Decoder, Codebook	Reconstruction
AR on VQVAE	Categorical Distribution (Dirac)	AR model	AR model	Generation
VA-π (Ours)	Teacher-forced Posterior via AR	AR model	AR model	Reconstruction + Generation

Table 6. Comparison of probabilistic formulations among VAE, VQVAE, VQVAE + AR, and the proposed VA-$\boldsymbol{\pi}$. Our method redefines the posterior through teacher-forced AR modeling, maintaining ELBO validity while enabling post-training alignment between the AR generator and the tokenizer.

form prior and the trained empirical posterior, it's challenging to generate realistic samples by decoding directly from the uniform prior. To bridge this gap, an AR model is subsequently trained to learn a more accurate prior over the token space, replacing the uniform assumption in VQVAE during sampling. This two-stage design inherently creates a gap between the AR model and the original VQVAE tokenizer.

While such gap is difficult to be resolved in the pretrained-stage due to the challenging optimization over non-differentiable discrete latent variable, we consider that it can be solved in the post-training settings. Given pretrained autoregressive model, we can reuse it as the prior and redefine the posterior to introduce sampling.

VA-$\boldsymbol{\pi}$ also represents the latent variable as a discrete token sequence. We redefine the posterior as follows: given a reference image, the encoder and quantizer are used to obtain a deterministic code, similar to the standard VQVAE formulation. However, unlike VQVAE, we further *teacher-force* this code into the AR model to compute a categorical distribution. This design keeps the ELBO theoretically valid, as the bound holds regardless of the specific choice of posterior, while offering two practical advantages. (1) Since the AR model is incorporated into the posterior, it receives direct supervision from the reconstruction term defined in pixel space. (2) It makes the KL regularization more interpretable: the KL divergence between the teacher-forcing distribution and the free-running distribution corresponds to the exposure bias, which can be reduced through next-token prediction under noisy ground-truth contexts, as shown in prior works [13, 14]. Redefining the posterior enables post-training of the AR model to better align with the tokenizer.

A.3. Details of Prior Regularization

To mitigate the inconsistency between the teacher-forced distribution (conditioned on a dataset) and the free-running distribution, we consider a regularization objective that directly addresses the *exposure bias* problem. While next-token prediction under noisy context has been shown to address exposure bias effectively in existing works [13, 62], we provide theoretical formulation below.

Next Token Prediction Regularization. We define the regularization objective as maximizing:

$$-\mathrm{KL}(q_{\phi,\theta}(\mathbf{x} \mid \mathbf{I}) \,\|\, \pi_\theta(\mathbf{x})) \quad \text{w.r.t. } \theta,$$

since the AR model used for teacher forcing is parameterized by the same model, given $\mathbf{x}^* \sim \mathcal{Q}(\mathcal{E}_\phi(\mathbf{I}))$, the objective is equivalent to minimizing:

$$\mathcal{L}_{\text{prior}}(\theta) = \mathrm{KL}(\pi_\theta(\mathbf{x} \mid \mathbf{x}^*) \,\|\, \pi_\theta(\mathbf{x}))\,. \tag{21}$$

where the AR model π_θ factorizes as:

$$\begin{aligned} \pi_\theta(\mathbf{x}) &= \prod_{t=1}^{N} \pi_\theta(x_t \mid x_{<t}), \\ \pi_\theta(\mathbf{x} \mid \mathbf{x}^*) &= \prod_{t=1}^{N} \pi_\theta(x_t \mid x^*_{<t}). \end{aligned} \tag{22}$$

For any AR laws $P(\mathbf{x}) = \prod_t P(x_t \mid x_{<t})$ and $Q(\mathbf{x}) = \prod_t Q(x_t \mid x_{<t})$, the chain rule for KL gives:

$$\mathrm{KL}(P\|Q) = \sum_{t=1}^{N} \mathbb{E}_{x_{<t}\sim P}\big[\mathrm{KL}\big(P(\cdot\,|\,x_{<t}) \,\|\, Q(\cdot\,|\,x_{<t})\big)\big]. \tag{23}$$

We also use the cross-entropy decomposition $\mathrm{KL}(p\|q) = \mathcal{H}(p,q) - \mathcal{H}(p)$, with $\mathcal{H}(p,q) := -\mathbb{E}_{y\sim p}[\log q(y)]$.

Applying (23) to $\pi_\theta(\mathbf{x} \mid \mathbf{x}^*)$ vs. $\pi_\theta(\mathbf{x})$,

$$\begin{aligned} \mathcal{L}_{\text{prior}}(\theta) &= \sum_{t=1}^{N} \mathbb{E}_{\mathbf{x}\sim\pi_\theta(\cdot|\mathbf{x}^*)}\Big[\mathrm{KL}\big(\pi_\theta(\cdot \mid x^*_{<t}) \,\|\, \pi_\theta(\cdot \mid x_{<t})\big)\Big] \\ &= \sum_{t=1}^{N} \mathbb{E}_{\mathbf{x}\sim\pi_\theta(\cdot|\mathbf{x}^*)}\Big[\mathcal{H}\big(\pi_\theta(\cdot \mid x^*_{<t}), \pi_\theta(\cdot \mid x_{<t})\big) \\ &\qquad - \mathcal{H}\big(\pi_\theta(\cdot \mid x^*_{<t})\big)\Big]. \end{aligned} \tag{24}$$

We make two assumptions:

A1 (*Teacher-forced calibration*) $\pi_\theta(\cdot \mid x^*_{<t}) = p^*(\cdot \mid x^*_{<t})$ for all $t, x^*_{<t}$, given π_θ is exactly pretrained on this.

A2 (*Prefix-matching corruption*) There exists a Markov kernel $K_\xi(\tilde{x}_{<t} \mid x^*_{<t})$ such that, when $\mathbf{x} \sim \pi_\theta(\cdot \mid \mathbf{x}^*)$,

$$p(x_{<t} \mid \mathbf{x}^*) \;=\; K_t(\cdot \mid x^*_{<t}) \quad \text{for all } t. \tag{25}$$

Using (25) to replace the expectation over $x_{<t}$ in (24) by

an expectation over $\tilde{x}_{<t} \sim K_t(\cdot \mid x^*_{<t})$, and applying A1,

$$\mathcal{L}_{\text{prior}}(\theta) = \sum_{t=1}^{N} \mathbb{E}_{\tilde{x}_{<t}\sim K_t(\cdot|x^*_{<t})} \mathbb{E}_{y\sim p^*(\cdot|x^*_{<t})} \big[-\log \pi_\theta(y \mid \tilde{x}_{<t}) \big] - \sum_{t=1}^{N} \mathcal{H}\big(p^*(\cdot \mid x^*_{<t})\big). \tag{26}$$

Let $C := \sum_t \mathcal{H}(p^*(\cdot \mid x^*_{<t}))$, which is independent of θ. Then:

$$\mathcal{L}_{\text{prior}}(\theta) = \underbrace{\sum_{t=1}^{N} \mathbb{E}_{\tilde{x}_{<t}\sim K_t} \mathbb{E}_{y\sim p^*(\cdot|x^*_{<t})} \big[-\log \pi_\theta(y \mid \tilde{x}_{<t}) \big]}_{\mathcal{L}_{\text{NTP-noisy}}(\theta)} - C. \tag{27}$$

Hence, minimizing the prior KL is equivalent up to a constant to minimizing the next-token prediction (NTP) loss under perturbed prefixes $\tilde{x}_{<t}$. While the assumption is not guaranteed in the experimental settings, we found it works empirically well by choosing proper corruption kernel.

Corruption Kernel Details. Following previous work [13], we use uniform noise to implement the corruption kernel. Given a discrete sequence $\mathbf{x}^* = (x^*_1, \ldots, x^*_N)$, we define a corruption kernel $K_\xi(\hat{\mathbf{x}} \mid \mathbf{x}^*)$ that introduces random perturbations with rate $\xi \in [0, 1]$. For each position $i \in \{1, \ldots, N\}$, we independently draw:

$$\hat{x}_i = \begin{cases} x^*_i, & \text{with probability } 1-\xi, \\ u_i, & \text{with probability } \xi, \end{cases} \tag{28}$$

where u_i is sampled uniformly from the token vocabulary $\mathcal{V} = \{1, \ldots, K\}$ excluding the ground-truth token, i.e. $u_i \sim \text{Unif}(\mathcal{V} \setminus \{x^*_i\})$. Equivalently, the conditional probability mass function is:

$$K_\xi(\hat{x}_i \mid x^*_i) = (1-\xi)\,\delta(\hat{x}_i = x^*_i) + \xi\,\frac{\mathbf{1}[\hat{x}_i \neq x^*_i]}{K-1}, \tag{29}$$

and the full sequence corruption factorizes as $K_\xi(\hat{\mathbf{x}} \mid \mathbf{x}^*) = \prod_{i=1}^{N} K_\xi(\hat{x}_i \mid x^*_i)$.

B. Tokenizer Training Objectives

The visual tokenizer is trained to map continuous image features into a compact set of discrete embeddings while maintaining perceptual reconstruction quality. As discussed in Sec. 3, the total objective in Eq. 2 consists of reconstruction and quantization terms:

$$\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}} + \lambda_{\text{q}}\mathcal{L}_{\text{q}}. \tag{30}$$

Pixel-wise reconstruction loss. We employ the mean squared error (MSE) between the original image $\mathbf{I}$ and its reconstruction $\hat{\mathbf{I}}$:

$$\mathcal{L}_{\text{MSE}} = \|\mathbf{I} - \hat{\mathbf{I}}\|_2^2. \tag{31}$$

Perceptual reconstruction loss. Following [50], a perceptual loss $\mathcal{L}_{\text{p}}$ compares high-level feature activations of $\mathbf{I}$ and $\hat{\mathbf{I}}$ extracted by a pretrained VGG network:

$$\mathcal{L}_{\text{p}} = \sum_{l} \|\phi_l(\mathbf{I}) - \phi_l(\hat{\mathbf{I}})\|_2^2, \tag{32}$$

where $\phi_l(\cdot)$ denotes features from the l-th layer.

Vector quantization loss. The quantization loss [18] encourages the encoder output $\mathbf{z}$ to commit to the closest embedding vector in the codebook $\mathcal{E} = \{e_k\}_{k=1}^K$:

$$\mathcal{L}_{\text{q}} = \|\text{sg}[\mathbf{z}] - e\|_2^2 + \beta\|\mathbf{z} - \text{sg}[e]\|_2^2, \tag{33}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and β controls the commitment strength. The first term updates the codebook entries to match encoder outputs, while the second prevents codebook collapse by penalizing large deviations of $\mathbf{z}$ from its assigned embedding.

Together, these terms ensure high-fidelity reconstruction, perceptual realism, and stable codebook learning. The resulting tokenizer provides discrete tokens that effectively balance compression and visual quality for subsequent autoregressive modeling.

C. Implementation Details

We first present the detailed formulation of baselines for mitigating discrepancy between decoding the AR-generated token sequences and the ground-truth image distribution by fine-tuning the AR model by STE algorithm C.1 or fine-tuning the tokenizer decoder C.2. Then, we present the implementation details of our VA-π on three experimental setting as in Tab. 8: LlamaGen for C2I/T2I, Janus-Pro for T2I.

C.1. Straight-Through Estimator Variants

Autoregressive (AR) image generators operate over discrete token sequences. Because the selection process (*e.g.*, $\arg\max$) is non-differentiable, we employ the *Straight-Through Estimator (STE)* [18] to provide a differentiable surrogate for the discrete sampling.

General Forward and Backward Pass. Given an image $\mathbf{I}$, the tokenizer encodes it into discrete token indices $\mathbf{x}^* = \mathcal{Q}(\mathcal{E}(\mathbf{I}))$. During training, the AR generator π_θ predicts the next-token logits $\mathbf{l}_t \in \mathbb{R}^V$ conditioned on $\mathbf{x}^*_{<t}$. To bridge the discrete and continuous spaces, we define a quantized representation $\mathbf{z}_q = \mathbf{y}_{st}^\top E$, where $E \in \mathbb{R}^{V\times D}$ is the frozen codebook. The surrogate one-hot vector $\mathbf{y}_{st}$ is computed as:

$$\mathbf{y}_{st} = \text{sg}[\mathbf{y}_{\text{hard}} - \mathbf{y}_{\text{soft}}] + \mathbf{y}_{\text{soft}}$$

Table 7. Ablation study on STE variants. Gumbel-Softmax provides a superior differentiable surrogate compared to Naive Softmax STE.

Method	Relaxation ($\mathbf{y}_{\text{soft}}$)	FID (↓)	IS (↑)
Naive STE	Standard Softmax	36.70	12.4
Gumbel STE	Gumbel-Softmax	**11.46**	**28.7**

where $\mathbf{y}_{\text{hard}} = \text{onehot}(\arg\max(\mathbf{l}_t))$, $\text{sg}[\cdot]$ is the stop-gradient operator, and $\mathbf{y}_{\text{soft}}$ is a continuous, differentiable relaxation. During the backward pass, the gradient flows through $\mathbf{y}_{\text{soft}}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{l}_t} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_q} \cdot \frac{\partial \mathbf{z}_q}{\partial \mathbf{y}_{st}} \cdot \frac{\partial \mathbf{y}_{st}}{\partial \mathbf{l}_t} \approx \frac{\partial \mathcal{L}}{\partial \mathbf{z}_q} \cdot E^\top \cdot \frac{\partial \mathbf{y}_{\text{soft}}}{\partial \mathbf{l}_t}$$

This ensures that each logit l_k is updated based on the dot product between the pixel-space gradient and its corresponding codebook entry. We investigate two distinct variants for constructing $\mathbf{y}_{\text{soft}}$.

Variant 1: Standard Softmax (Naive STE). In this deterministic variant, $\mathbf{y}_{\text{soft}}$ is obtained by applying a standard Softmax function with a temperature parameter τ:

$$y_{\text{soft}}^{(k)} = \frac{\exp(l_{t,k}/\tau)}{\sum_{j=1}^{V} \exp(l_{t,j}/\tau)}$$

While mathematically straightforward, this Naive STE relies entirely on the raw logits. The deterministic nature of the Softmax restricts exploration in the discrete latent space, often leading to poor gradient approximation and a higher risk of mode collapse during AR fine-tuning.

Variant 2: Gumbel-Softmax STE. To introduce stochasticity and achieve smoother gradient flows, we incorporate the Gumbel-Softmax trick [54]. We add i.i.d. noise $g_k \sim \text{Gumbel}(0, 1)$ to each logit before applying the Softmax function:

$$y_{\text{soft}}^{(k)} = \frac{\exp((l_{t,k} + g_k)/\tau)}{\sum_{j=1}^{V} \exp((l_{t,j} + g_j)/\tau)}$$

where the Gumbel noise is sampled via $g_k = -\log(-\log(u_k))$ with $u_k \sim \text{Uniform}(0, 1)$. The injected noise acts as a regularizer, enabling a stochastic gradient estimator that effectively prevents the generator from getting stuck in local optima.

Comparison and Empirical Results. To evaluate the impact of the relaxation strategy, we conduct an ablation study comparing the Naive Softmax STE and the Gumbel-Softmax STE. As shown in Tab. 7, the choice of $\mathbf{y}_{\text{soft}}$ is critical.

The Gumbel-Softmax variant significantly outperforms the Naive STE across image quality metrics (FID 11.46 vs. 36.70). The deterministic Softmax struggles to provide informative gradients through the $\arg\max$ bottleneck, resulting in degraded generation quality. However, despite its effectiveness, STE-based fine-tuning still requires significantly higher computational costs because gradients must be propagated through the entire vocabulary at each step. This limitation motivates our proposed VA-π, which avoids full gradient propagation through the codebook by operating directly on the ground-truth path.

C.2. Tokenizer Post-training

A tokenizer trained only on ground-truth image reconstructions may not fully account for the distribution shift induced by AR-generated token sequences. From the tokenizer's perspective, enhancing robustness to such off-manifold or suboptimal token sequences provides an additional means to reduce the mismatch between AR-generated token distributions and the underlying image distribution.

Formulation. Recall that the tokenizer consists of an encoder $\mathcal{E}$, a quantizer $\mathcal{Q}$, and a decoder $\mathcal{D}_\phi$. Given an image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, the tokenizer produces its latent representation and discrete token sequence:

$$\mathbf{z} = \mathcal{E}(\mathbf{I}), \qquad \mathbf{x}^* = \mathcal{Q}(\mathbf{z}), \tag{34}$$

where $\mathbf{z} \in \mathbb{R}^{C\times N}$ and $\mathbf{x}^* \in \{1, \ldots, K\}^N$. The sequence $\mathbf{x}^*$ serves as the teacher-forcing target for the AR model, whose parameters are trained by maximizing:

$$\theta = \arg\max_\theta \sum_{i=1}^{N} \log \pi_\theta(x_i^* \mid \mathbf{x}_{1:i-1}^*). \tag{35}$$

During tokenizer post-training, we keep the AR model π_θ, the quantizer $\mathcal{Q}$, and its codebook fixed. Given a teacher-forcing sequence $\mathbf{x}^*$, the AR model generates its own token predictions autoregressively:

$$x_i \sim \pi_\theta(\cdot \mid \mathbf{x}_{1:i-1}^*), \qquad \mathbf{x} = (x_1, \ldots, x_N), \tag{36}$$

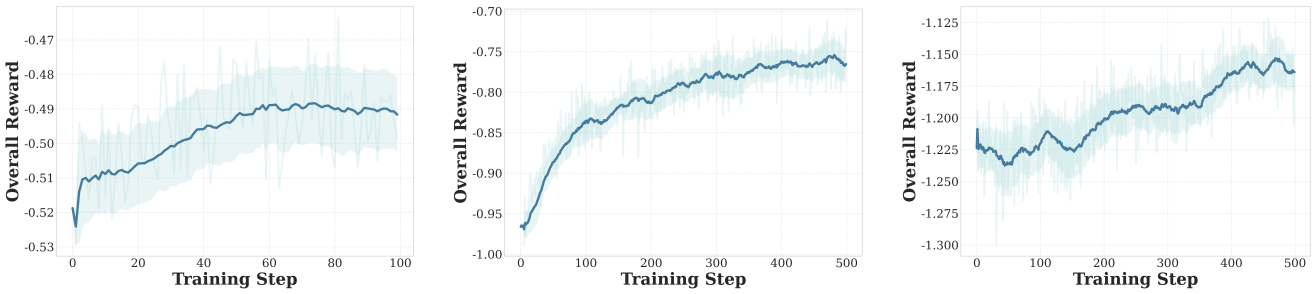
which reflect the model's free-running token distribution conditioned on $\mathbf{x}^*$. The AR-generated sequences $\mathbf{x}$ are decoded by the tokenizer decoder:

$$\hat{\mathbf{I}} = \mathcal{D}_\phi(\mathbf{x}), \tag{37}$$

and the decoder parameters ϕ are updated to improve the reconstruction fidelity of AR-generated tokens. Because the quantizer $\mathcal{Q}$ and the AR model π_θ remain frozen, the teacher-forcing token distribution $\mathbf{x}^*$ is preserved, ensuring compatibility with the pretrained AR model while enhancing robustness to its sampled token sequences.

Objective. Unlike the full tokenizer objective in Eq. 2, the post-training objective excludes the quantization loss and focuses solely on reconstruction fidelity:

$$\mathcal{L}_{\text{PT}} = \|\hat{\mathbf{I}} - \mathbf{I}\|_2^2 + \lambda_{\text{p}}\, \mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{I}}, \mathbf{I}). \tag{38}$$



(a) (b) (c)

Figure 5. **Learning curves of our reinforcement-learning framework VA-π across three model scenarios.** (a) C2I (LlamaGen-XXL, 100 steps), (b) T2I (LlamaGen-XL, 500 steps), and (c) T2I (Janus-Pro 1B, 500 steps).

This updates only the reconstruction pathway without altering the discrete token space.

The improved tokenizer thus serves as a drop-in replacement that yields higher-fidelity decoding for both ground-truth and AR-generated token sequences without inducing any distributional mismatch.

C.3. Implementation Details of VA-π

We summarize the hyperparameters used for RL fine-tuning across all models and tasks in Tab. 8. For both C2I and T2I experiments with LlamaGen, we adopt the same optimizer configuration—AdamW with a learning rate of 1×10^{-6}, weight decay of 1×10^{-4}, `bf16` mixed precision, and a global batch size of 128. For Janus-Pro 1B, we use a smaller learning rate (1×10^{-7}) due to its increased sensitivity during RL updates. All models are trained with group size G=8, advantage clipping at 5.0, a classifier-free guidance scale of 1.0, and a maximum gradient norm of 1.0.

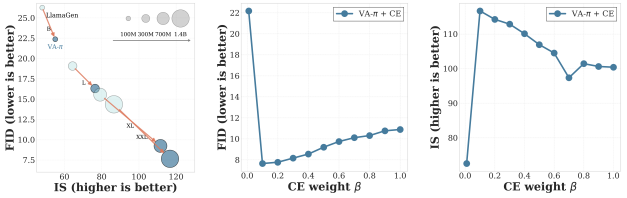
We fine-tune LlamaGen-XXL for 100 steps on C2I and LlamaGen-XL for 200 steps on T2I, while Janus-Pro 1B is trained for 100 steps given its heavier multimodal architecture. Image resolution is set to 384×384 for all LlamaGen models and 256×256 for Janus-Pro 1B.

D. Additional Quantitative Results

D.1. Ablation on the Alignment Design

In this subsection, we provide additional ablations on two important design choices in VA-π: whether classifier-free guidance (CFG) should be enabled during alignment training, and how sensitive the method is to the weight of the auxiliary cross-entropy term. These experiments complement the main paper by showing that our default setting is not only empirically effective but also stable across a reasonably broad range of choices.

We first study the effect of enabling CFG during alignment training. While this choice is in principle compatible with our framework, our default setting does not apply CFG in training. This is because VA-π relies on teacher-forced samples for constructing pixel-aware supervision, whereas CFG primarily affects free-running decoding at inference time. As a result, applying CFG during training can introduce inconsistency and slightly destabilize optimization.



(a) (b) (c)

Figure 6. **Stability across model scales and hyperparameter sensitivity.** (a) VA-π shows consistent performance gains across four model scales (B, L, XL, XXL) and two resolutions ($16 \times 16, 24 \times 24$). (b-c) Evaluation of optimization stability over more CE weight β, demonstrating a robust peak performance region.

We next extend the sensitivity analysis of the auxiliary CE weight β to a denser range from 0.0 to 1.0 with a step size of 0.1. This experiment is intended to verify that the improvement of VA-π does not rely on a narrowly tuned coefficient. As shown in the following figures, introducing the CE term is crucial when moving from $\beta = 0$ to a small positive value, while the performance remains strong over a broad plateau once the regularization is activated.

Specifically, FID improves substantially when moving from $\beta = 0.0$ to $\beta = 0.1$, and the performance remains stable for larger values of β. This trend indicates that the auxiliary CE term is essential for stable optimization, while the method itself is not overly sensitive to the precise coefficient within a reasonable range.

D.2. Stability Across Model Scales

To examine whether the benefit of VA-π is specific to a single model size, we further evaluate it across generators of different scales. This analysis is important because a lightweight post-training method should ideally provide

Table 8. **Hyperparameters used for RL fine-tuning across three model settings.** LlamaGen-XXL (C2I) and LlamaGen-XL (T2I) use consistent optimization settings, while Janus-Pro 1B adopts a smaller learning rate and lower resolution due to its multimodal architecture and training stability considerations.

Name	LlamaGen for C2I	LlamaGen for T2I	Janus-Pro 1B for T2I
Learning Rate	1e-6	1e-6	1e-7
Weight Decay	1e-4	1e-4	1e-4
Mixed Precision	bf16	bf16	bf16
Beta β	0.1	0.1	0.1
Group Size G	8	8	8
Max Advantage Clip	5.0	5.0	5.0
Classifier-Free Guidance Scale	1.0	1.0	1.0
Max Gradient Norm	1.0	1.0	1.0
Total Batchsize per Step	128	128	128
Training Steps	100	200	100
Image Resolution $h \times w$	384×384	256×256	384×384
Contextual Noise ξ	0	0.5	0.95

Table 9. **Representative prompts used for qualitative comparison on GenEval tasks.**

Task	Prompt	
Attribute Binding	"a photo of a brown laptop and a white bicycle" "a photo of a blue baseball glove and a black pizza"	"a photo of a black cow and an orange donut" "a photo of a purple chair and a brown surfboard"
Counting	"a photo of two cups" "a photo of three wine glasses"	"a photo of three cell phones" "a photo of four birds"
Position	"a photo of a sports ball right of a handbag" "a photo of a skateboard right of a fire hydrant"	"a photo of a handbag above a sheep" "a photo of a computer keyboard right of skis"
Two-Object Combination	"a photo of a frisbee and a bottle" "a photo of a skateboard and a baseball glove"	"a photo of a book and a tv" "a photo of a chair and a tv"

Table 10. **Ablation of CFG during the alignment phase.** Training without CFG achieves better FID and IS.

Training Setting	FID ↓	IS ↑
VA-π w/ CFG Training	3.56	260.31
VA-π w/o CFG Training (Ours)	**2.28**	**273.53**

consistent gains without requiring architecture-specific tuning. The following results show that the advantage of VA-π is preserved as the backbone scale increases.

Fig. 6a summarizes the performance of baseline AR generators and their VA-π-aligned counterparts across different model sizes. We observe a consistent improvement trend in both image quality and semantic fidelity, indicating that the proposed alignment objective scales smoothly with model capacity rather than only benefiting a narrow operating regime.

The consistent gain across scales suggests that the improvement brought by VA-π does not depend on a single architecture size or a specially tuned operating regime. Instead, the pixel-aware alignment objective appears to remain effective as the generator capacity grows.

E. Additional Qualitative Results

E.1. Visualization of Learning Curves

We visualize the learning curves during reinforcement learning fine-tuning, showing how the reward evolves as training progresses. As illustrated in Fig. 5, the reward steadily increases across all settings, indicating that the policy consistently improves under our VA-π framework. These curves confirm that our reinforcement learning framework enables robust policy improvement across generators of varying scales and modalities.

E.2. Class-to-Image Generation

We present additional qualitative comparisons on the class-to-image (C2I) generation task, showcasing examples from various ImageNet [19] classes. Each comparison visualizes generations from the baseline LlamaGen-XXL [20] and our VA-π, which applies reinforcement learning post-training

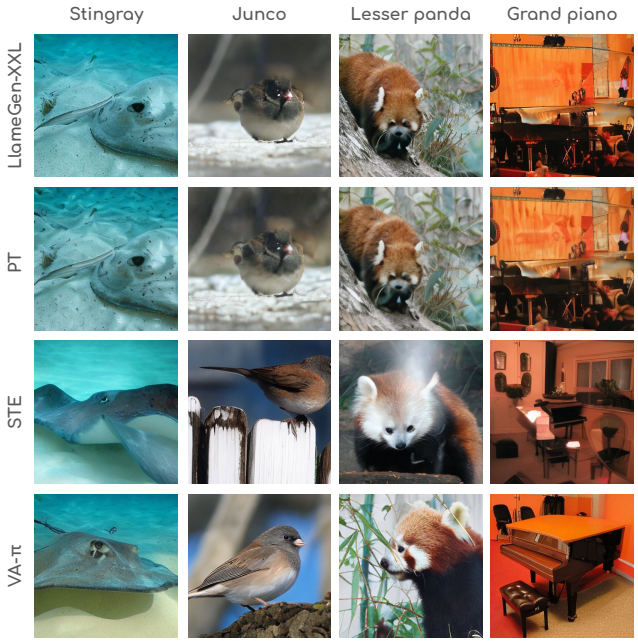


Figure 7. **Qualitative comparison of C2I generation among LlamaGen-XXL [20], post-train tokenizer (PT), STE based post-train AR and VA-π on the ImageNet-1k [19] classes.** Both models use a CFG scale of 1.0. VA-π shows better semantic alignment and image quality, demonstrating that pixel-space alignment encourages realistic generations.

Figure 8. **Observation on Post-Train Tokenizer.** Post-training the tokenizer decoder produces smoother and less detailed textures, despite preserving global structure. This over-smoothing effect explains why extended decoder fine-tuning leads to worse FID (14.36 $\rightarrow$ 22.99) and IS (86.55 $\rightarrow$ 72.49), highlighting the inherent limitation of decoder-only fine-tuning.

on LlamaGen-XXL. All samples are generated under identical decoding configurations with CFG scale = 1.0, temperature = 1.0, top-k = 0, and top-p = 1.0.

We also present additional qualitative comparisons on fine-tuning methods like STE [12] and post-train tokenizer as shown in Fig. 7.

We further analyze the impact of long-term tokenizer decoder post-training. Surprisingly, although reconstruction loss steadily decreases during training, both FID and IS consistently worsen. This degradation arises because the decoder, trained with teacher-forced ground-truth tokens, gradually learns an overly smooth reconstruction mapping as shown in Fig. 8. It becomes tolerant to token inaccuracies, suppresses high-frequency textures, and specializes in cleaning input tokens that do not match the noisy AR tokens produced during inference. As a result, images retain coarse structure but lose sharpness and perceptual richness.

These observations highlight the inherent limitations of decoder-only fine-tuning: it cannot correct off-manifold token sequences, amplifies the train–inference mismatch, and ultimately smooths away meaningful details. This reinforces the central design choice of our method—fine-tuning the AR generator itself is necessary to align the generated token distribution with the ground-truth image manifold.

Post-training the tokenizer (PT) preserves coarse structure but produces overly smooth textures, while STE-based AR fine-tuning partially improves details yet still suffers from off-manifold token transitions. This is because STE only updates the likelihood of teacher-forced tokens and cannot adjust the AR model's sampling behavior. In contrast, our VA-π optimizes pixel-space rewards for sampled token sequences, yielding sharper textures, more coherent semantics, and overall more realistic generations. These qualitative results highlight that pixel-level reward alignment is essential for correcting sampling errors that STE alone cannot address.

Overall, VA-π produces images that are more consistent with class semantics and exhibit improved structural fidelity and perceptual realism compared to the pre-trained LlamaGen-XXL baseline.

E.3. Text-to-Image Generation

We present additional qualitative comparisons on the text-to-image (T2I) generation task, showcasing examples from various prompts from the the GenEval benchmark [55]. Especially, we present images generated from complex tasks: attribute binding, counting, position, and two-object combination. Each comparison visualizes generations from the unified multi-modal baseline Jans-Pro 1B [48] and our VA-π, which applies reinforcement learning post-training on Jans-Pro 1B. All samples are generated under identical decoding configurations with CFG scale = 5.0, temperature = 1.0, top-k = 0, and top-p = 1.0.

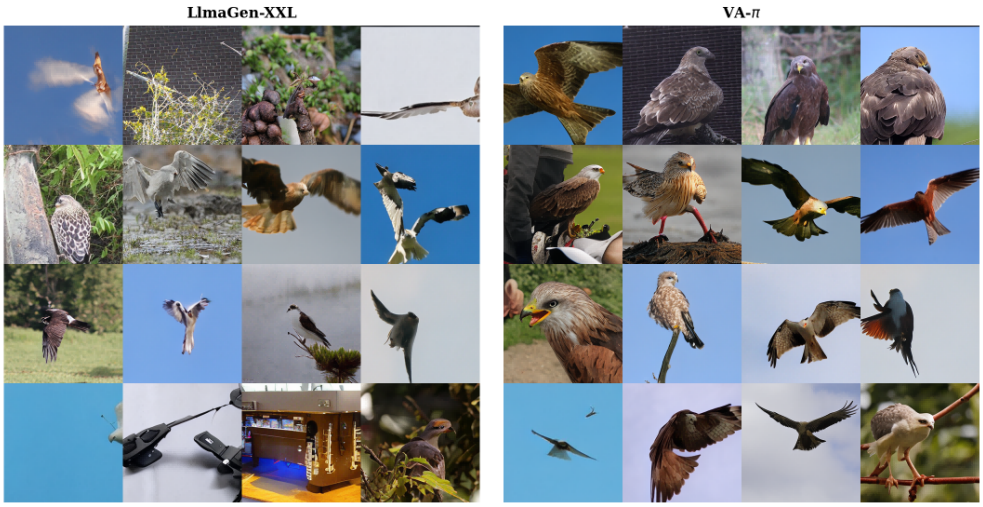


Figure 9. Qualitative comparison on the `kite` class.

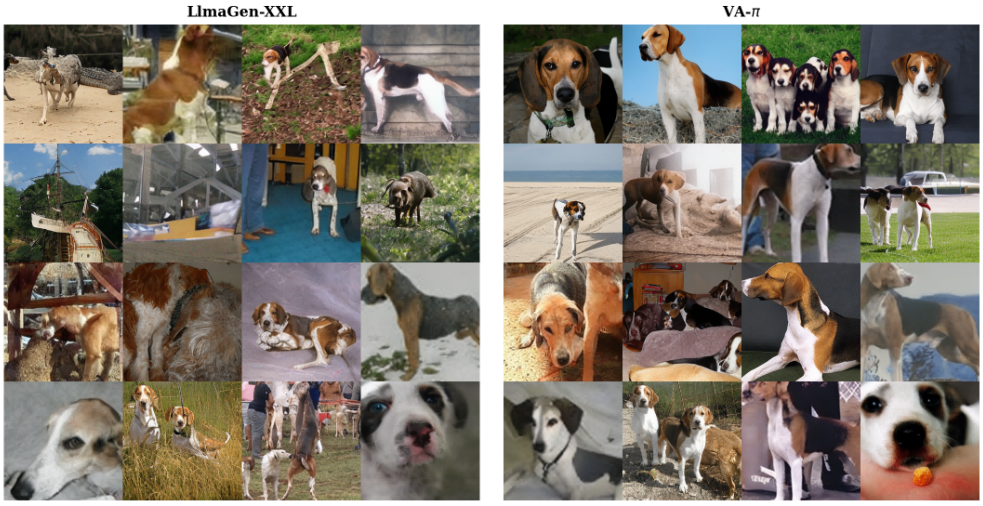


Figure 10. Qualitative comparison on the `English foxhound` class.

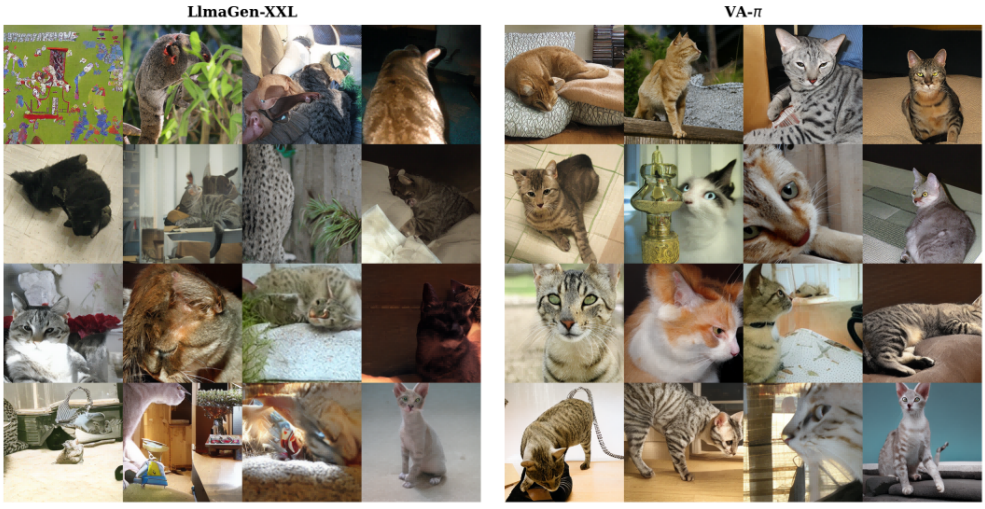


Figure 11. Qualitative comparison on the `Egyptian cat` class.

Figure 12. Qualitative comparison on the `dome` class.

Figure 13. Qualitative comparison on the `thresher / thrasher / threshing machine` class.

Figure 14. Qualitative comparison on the `cheese burger` class.

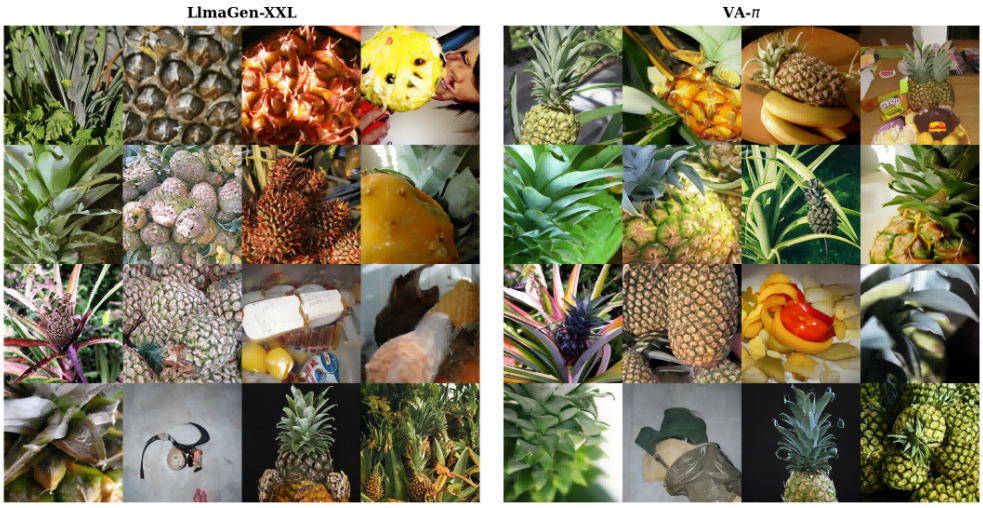


Figure 15. Qualitative comparison on the `pineapple / ananas` class.

Figure 16. Qualitative comparison on the `bolete` class.

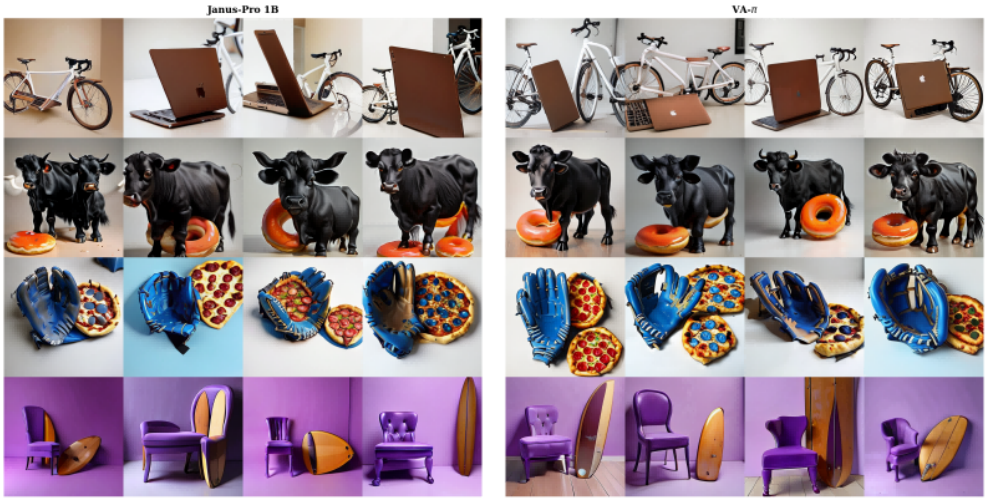


Figure 17. Qualitative comparison on the `attribute binding` task.

Figure 18. Qualitative comparison on the `counting` task.

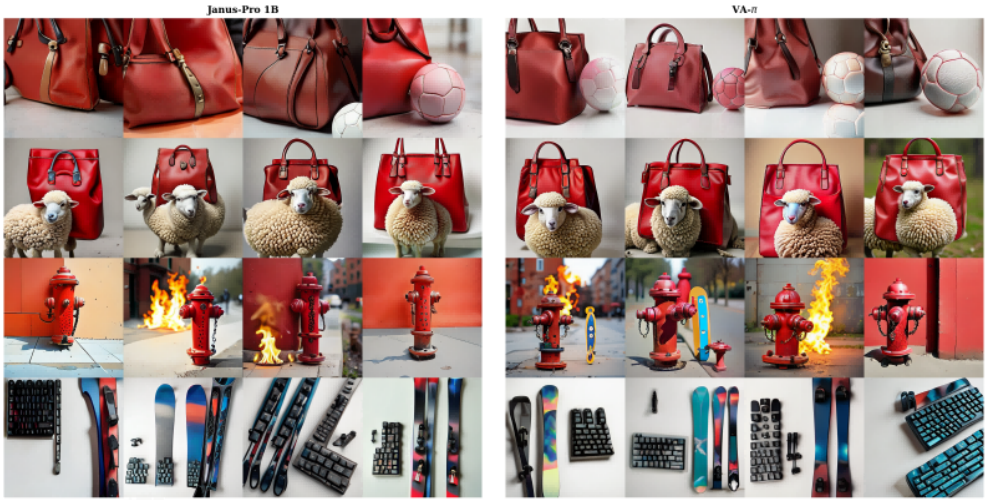


Figure 19. Qualitative comparison on the `position` task.

Figure 20. Qualitative comparison on the `two-object combination` task.