LEARNING DISCRETE AUTOREGRESSIVE PRIORS WITH WASSERSTEIN GRADIENT FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete image generation commonly uses two-stage training: a tokenizer is first trained for reconstruction, and a prior is then fitted to the tokens from the frozen tokenizer. This decoupling leaves the tokenizer unaware of the model that will later generate its tokens. As a result, the learned tokens may preserve image information well but still be difficult for an autoregressive (AR) prior to predict from left to right. We analyze this mismatch using Tripartite Variational Consistency (TVC), which decomposes latent-variable learning into three consistency conditions: conditional-likelihood consistency, prior consistency, and posterior consistency. TVC shows that two-stage training preserves the reconstruction side but leaves prior consistency outside the tokenizer objective: the overall token distribution is fixed before the AR prior participates in training. Motivated by this view, we add a distribution-level prior-matching signal during tokenizer training, while keeping the reconstruction objective unchanged. We optimize this signal with a Wasserstein-gradient-flow update. For hard categorical tokens, we approximate the update with a token-level contrast between an auxiliary AR model that tracks the tokenizer's current token distribution and a frozen AR reference. Computing this tokenizer update requires only forward passes through the two AR models; the proxy uses its own cross-entropy loss to track the current tokens. After tokenizer training, a separate AR prior is trained for generation. The resulting tokenizer, **wAR-Tok**, reduces AR loss and improves generation FID on CIFAR-10 and ImageNet at comparable reconstruction quality.

1 INTRODUCTION

Latent generative models first compress an image into a shorter representation, and then learn a generative model over that representation (Kingma & Welling, 2014). This separation is used in Latent Diffusion Models (Rombach et al., 2022), Masked Image Modeling (Chang et al., 2022), and discrete image tokenizers. A common recipe is two-stage training (Esser et al., 2021): the tokenizer is first trained to reconstruct images, and a prior is then trained on the frozen token sequences. This recipe makes the tokenizer good at reconstruction, but it does not ask whether the resulting tokens are easy for the second-stage prior to model. In 1D tokenization (Yu et al., 2024), this issue becomes visible with autoregressive (AR) priors (Sun et al., 2024): the same tokens can reconstruct images well but remain difficult to predict from left to right.

Recent AR-alignment methods try to make the learned tokens more sequential. Tail dropout (Zeghidour et al., 2021; Bachmann et al., 2025) trains the decoder to reconstruct from prefixes, encouraging earlier tokens to carry useful information. Flow-matching decoders such as FlowMo (Sargent et al., 2025) reconstruct images through iterative refinement. SelfTok (Wang et al., 2025) ties the prefix length to the diffusion timestep. These methods can improve the ordering of information in the token sequence, but the supervision still comes through the decoder. The AR prior itself is not used to train the tokenizer, so its modeling difficulty is only addressed indirectly.

We use Tripartite Variational Consistency (TVC, §3) to make this missing link explicit. TVC views latent-variable learning as the consistency of three parts: the conditional likelihood, the prior, and the posterior. This perspective separates two different failure modes. ELBO-style training forces each input's latent distribution to look like the prior, which can remove the information that the latent should carry. Two-stage training avoids this collapse by training the tokenizer only for reconstruction,

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

but then leaves the entire burden of prior matching to the second stage. In other words, the tokenizer fixes a latent distribution before the AR prior ever participates in training.

This analysis points to a more direct objective. The tokenizer should still be trained for reconstruction, but it should also receive a signal that makes its overall token distribution closer to what the AR prior can model. We call this term Distributional Prior Divergence (DPD). Unlike the instance-wise constraint in ELBO, DPD only matches the aggregate distribution of tokens. Unlike standard two-stage training, it sends this prior-matching signal back to the tokenizer. Existing aggregate-matching methods such as WAE-MMD (Tolstikhin et al., 2017), WAE-GAN/AAE (Makhzani et al., 2015), and InfoVAE (Zhao et al., 2017) pursue a related goal in continuous latent spaces, but they do not directly apply to hard discrete tokenizers. Direct AR-loss training also guides the tokenizer through a trainable prior (Wang et al., 2024), with a reconstruction–generation trade-off that depends on the loss weight (Appendix B).

The remaining question is how to optimize this distribution-level mismatch for discrete tokens. We do this with a Wasserstein-gradient-flow update. Intuitively, the update compares two token-level predictions: one from a proxy model that tracks the tokenizer's current token distribution, and one from a frozen AR reference. If the proxy assigns higher probability to a token than the reference does, the tokenizer is pushed away from that choice; if the reference assigns higher probability, the tokenizer is pulled toward it. This tokenizer update needs only forward passes through the two AR models; the proxy uses its own cross-entropy loss. A separate generation prior is then trained on the frozen tokenizer. We call the method **WGF-DPD**, and the tokenizer trained with it **wAR-Tok**.

On CIFAR-10, under matched reconstruction quality (rFID ≈ 5.0), wAR-Tok improves gFID from 17.60 to 14.50 relative to the strongest baseline in our controlled setting and reduces AR generation loss from 0.78 to 0.42. Under matched training iterations, it reaches gFID 11.30 and gLoss 0.34. On ImageNet 256×256, using the same encoder–decoder architecture and AR prior as the baseline, wAR-Tok reduces gFID from 6.32 to 5.42 and evaluation AR loss from 7.89 to 7.07, with rFID changing from 2.05 to 2.35.

Our contributions are a TVC analysis of prior matching, a distribution-level objective for tokenizer training, and a WGF-derived update for hard discrete tokens. Controlled experiments evaluate the resulting reconstruction–generation trade-off.

2 PRELIMINARIES

We consider the problem of learning a generative model for high-dimensional data $\boldsymbol{x} \in \mathcal{X}$ drawn from an unknown data distribution $p_{\text{data}}(\boldsymbol{x})$. Latent Variable Models (LVMs) address this by introducing an unobserved latent variable $\boldsymbol{z} \in \mathcal{Z}$, positing that the complex data distribution arises from a structured joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z})$. The marginal likelihood of the model is obtained by integrating over the latent space:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\, p_\theta(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z},$$

where $p_\theta(\boldsymbol{z})$ is the prior and $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ is the conditional likelihood (or decoder). The objective of an LVM is to optimize θ such that the model marginal $p_\theta(\boldsymbol{x})$ approximates the empirical data distribution $p_{\text{data}}(\boldsymbol{x})$ as closely as possible.

Tokenizer and AR models. The encoder E_ϕ maps an image to features, a quantizer with codebook $\boldsymbol{C}$ maps them to discrete tokens $\boldsymbol{z}$, and the decoder D_ξ reconstructs the image from their codebook embeddings. We use $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ for the tokenizer distribution, with codebook dependence left implicit, and q_ψ for the online proxy. We use p_θ for the AR model: a frozen reference in Stage I and a separately trained generation prior in Stage II. The generic LVM notation groups decoder and prior parameters under θ; the method and Algorithm 1 use ξ for the decoder.

2.1 Evidence Lower Bound and Two-Stage Training

Given a dataset X, the Evidence Lower Bound (ELBO) (Kingma & Welling, 2014; Rezende et al., 2014) is

$$\sum_{\boldsymbol{x}\in X} \log p_\theta(\boldsymbol{x}) \geq \sum_{\boldsymbol{x}} \Big(\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})} \big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \big] - D_{\mathrm{KL}} \big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_\theta(\boldsymbol{z}) \big) \Big).$$

Let $q_\phi(\boldsymbol{z}) = \int p_{\text{data}}(\boldsymbol{x})\, q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ denote the aggregate posterior. Expanding the KL term in the ELBO yields

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[D_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\|\, p_\theta(\boldsymbol{z}))] = H_q(\boldsymbol{Z}) - H_q(\boldsymbol{Z} \mid \boldsymbol{X}) + D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z}))\,.$$

Under deterministic coding, $H_q(\boldsymbol{Z} \mid \boldsymbol{X}) = 0$, and the data-averaged objective can be further written as

$$\begin{aligned} &\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim q_\phi(\boldsymbol{x},\boldsymbol{z})} \big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \big] - \big(H_q(\boldsymbol{Z}) - H_q(\boldsymbol{Z} \mid \boldsymbol{X}) \big) - D_{\mathrm{KL}} \big(q_\phi(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z}) \big) \\ &= \underbrace{\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim q_\phi(\boldsymbol{x},\boldsymbol{z})} \big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \big]}_{\text{reconstruction}} + \underbrace{\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim q_\phi(\boldsymbol{x},\boldsymbol{z})} \big[\log p_\theta(\boldsymbol{z}) \big]}_{\text{prior mapping}}. \end{aligned}$$

Modern discrete generative models can be viewed as a *two-stage* training under this objective (van den Oord et al., 2017; Esser et al., 2021): the first stage optimizes the reconstruction term, while the second stage fits $q_\phi(\boldsymbol{z})$.

2.2 Autoregressive Modeling

Standard AR approaches (van den Oord et al., 2016) model the aggregate posterior $q_\phi(\boldsymbol{z})$ by maximizing the log-likelihood of the latent codes $\boldsymbol{z} = (z_1, \ldots, z_n)$ under a parametric prior $p_\theta(\boldsymbol{z})$. This is typically optimized via cross-entropy with teacher forcing (Williams & Zipser, 1989):

$$\mathcal{L}_{\text{AR}}(\theta) = -\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})} \left[\frac{1}{n} \sum_{t=1}^{n} \log p_\theta(z_t \mid \boldsymbol{z}_{<t}) \right],$$

where $\boldsymbol{z}_{<t} \triangleq (z_1, \ldots, z_{t-1})$ denotes the history of previous tokens.

2.3 Existing AR-Alignment Techniques

Three representative techniques recur across recent work: **Tail Dropout**, **Diffusion Decoder**, and **Tied Timesteps**. As a controlled baseline we also consider **1D Tokenization**, which removes spatial biases but does not explicitly target AR alignment.

1D Tokenization (Yu et al., 2024) feeds learnable holistic-token queries alongside image-patch embeddings to the encoder, yielding latent representations decoupled from explicit 2D structure.

Tail Dropout, first explored in audio codecs (Zeghidour et al., 2021) and adapted to vision (Bachmann et al., 2025), randomly discards a suffix of the token sequence during training and requires the decoder to reconstruct from any prefix:

$$\mathcal{L}_{\text{tail}} = -\mathbb{E}_k \big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}_{1:k}) \big].$$

Here k is sampled from the training truncation distribution, and $k = 0$ denotes an empty prefix.

Flow-Matching Decoder. FlowMo (Sargent et al., 2025) reconstructs images through iterative refinement. In our setup, this changes the image decoder; the token prior remains autoregressive.

Tied Timesteps (Wang et al., 2025) explicitly couples the tail-dropout ratio with the diffusion timestep, binding the discrete prefix length to a noise level. We discuss these methods further in Sec. 6.

3 TRIPARTITE VARIATIONAL CONSISTENCY

TVC provides a common view of reconstruction and prior matching. We define the data-anchored joint $q_\phi(\boldsymbol{x}, \boldsymbol{z}) = p_{\text{data}}(\boldsymbol{x}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and its aggregate posterior $q_\phi(\boldsymbol{z}) = \int p_{\text{data}}(\boldsymbol{x}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$. Comparing this joint with the model joint separates three consistency conditions.

Theorem 3.1 (Tripartite Variational Consistency)**.** *Where the density ratios are defined, Bayes' rule gives the following decomposition:*

$$\log p_\theta(\boldsymbol{x}) = \underbrace{\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})}{q_\phi(\boldsymbol{x} \mid \boldsymbol{z})}}_{\textit{Likelihood Consistency}} + \underbrace{\log \frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z})}}_{\textit{Prior Consistency}} - \underbrace{\log \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}}_{\textit{Posterior Consistency}} + \log p_{\textit{data}}(\boldsymbol{x}) \tag{1}$$

Satisfying these three consistency conditions gives $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$. This view identifies which conditions each training objective addresses. Proofs are in Appendix D.

3.1 CONSISTENCY PAIRS

Corollary 3.2 (Constraint Redundancy)**.** *Under the conditions in Appendix D, satisfying an admissible pair of consistency conditions implies the third and gives $p_\theta(\boldsymbol{x}) = p_{\textit{data}}(\boldsymbol{x})$.*

Case 1: Likelihood–Prior Matching matches $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ with $q_\phi(\boldsymbol{x} \mid \boldsymbol{z})$ and $p_\theta(\boldsymbol{z})$ with $q_\phi(\boldsymbol{z})$. Their product then matches the data-anchored joint, giving posterior consistency. **Case 2: Likelihood–Posterior Matching** instead matches the two conditionals. **Case 3: Posterior–Prior Matching** matches the posterior and prior, with the completeness condition stated in the appendix.

We use Case 1 because it preserves the existing reconstruction objective and adds a tractable prior-matching update. Cases 2 and 3 suggest future objectives that match the model posterior directly, but require a practical way to estimate that posterior. Appendix D.6 discusses these choices and gives an example objective.

3.2 ELBO AND TWO-STAGE TRAINING THROUGH TVC

Both approaches address likelihood and prior matching, but couple them differently. ELBO uses an instance-wise prior penalty, which also penalizes the information retained about the input. Two-stage training first learns reconstruction and then fits a prior, leaving prior matching outside tokenizer training. Aggregate-matching objectives such as WAE, AAE, and InfoVAE (Tolstikhin et al., 2017; Makhzani et al., 2015; Zhao et al., 2017) relax the instance-wise penalty. Our goal is to make this distribution-level signal practical for hard discrete tokenizers. The corresponding derivations are in Appendix D.2–D.4.

4 PRIOR MATCHING FOR DISCRETE TOKENIZERS

4.1 SUBOPTIMAL REPRESENTATIONS OF ELBO AND TWO-STAGE TRAINING

The limitations of both paradigms stem from their treatment of Prior Consistency. Under TVC, the ELBO can be interpreted as enforcing prior consistency through a strict **instance-wise** constraint. Specifically, satisfying $q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = p_\theta(\boldsymbol{z})$ for all $\boldsymbol{x} \sim p_{\text{data}}$ implies the alignment of the marginal distributions: $q_\phi(\boldsymbol{z}) = \int p_{\text{data}}(\boldsymbol{x}) \, q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = p_\theta(\boldsymbol{z})$. This condition is restrictive: strictly equating the conditional posterior to the prior implies that $\boldsymbol{z}$ is statistically independent of $\boldsymbol{x}$, causing the mutual information to vanish: $I(\boldsymbol{X}; \boldsymbol{Z}) = H_q(\boldsymbol{Z}) - H_q(\boldsymbol{Z} \mid \boldsymbol{X}) = 0$. This is the phenomenon of posterior collapse (Bowman et al., 2016; Lucas et al., 2019). Instance-wise alignment is a *sufficient* but not *necessary* condition for prior consistency. Prior consistency requires only that the *aggregate* posterior $q_\phi(\boldsymbol{z})$ matches the prior $p_\theta(\boldsymbol{z})$, which lets the encoder retain dependence on the input.

Two-stage methods take the opposite path. They first optimize the encoder $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ solely for reconstruction fidelity, leaving the aggregate posterior $q_\phi(\boldsymbol{z})$ unconstrained. The entire alignment burden then falls on the second stage. If the induced latent structure is statistically misaligned with the inductive bias of the AR prior (lacking causal ordering or containing high-frequency artifacts), fitting $p_\theta(\boldsymbol{z})$ becomes inefficient, and excessive parameter scaling is often needed to compensate.

Algorithm 1 wAR-Tok training.

Input: Tokenizer $(E_\phi, \boldsymbol{C}, D_\xi)$, proxy q_ψ, AR model p_θ, weight λ.

Stage I: Train the tokenizer.

Keep the AR reference p_θ frozen.
for each training batch $\boldsymbol{x}$ **do**
  $\boldsymbol{u} \leftarrow E_\phi(\boldsymbol{x})$; $\boldsymbol{h} \leftarrow \mathrm{softmax}(\boldsymbol{\ell}(\boldsymbol{u}, \boldsymbol{C})/\tau_{\mathrm{ST}})$
  $\boldsymbol{z} \leftarrow \mathrm{onehot}(\arg\max_k h_k)$; $\boldsymbol{z}_{\mathrm{ST}} \leftarrow \boldsymbol{h} + \mathrm{sg}(\boldsymbol{z} - \boldsymbol{h})$
  $\hat{\boldsymbol{x}} \leftarrow D_\xi(\boldsymbol{C}\boldsymbol{z})$; compute $\mathcal{L}_{\mathrm{rec}}$ with standard VQ STE
  $\boldsymbol{g}_t \leftarrow \mathrm{sg}(\log q_\psi(\cdot \mid \boldsymbol{z}_{<t}) - \log p_\theta(\cdot \mid \boldsymbol{z}_{<t}))$ for all t
  $\mathcal{L}_{\mathrm{align}} \leftarrow \frac{1}{n}\sum_t \langle \boldsymbol{z}_{\mathrm{ST},t}, \boldsymbol{g}_t \rangle$
  $\mathcal{L}_\psi \leftarrow -\frac{1}{n}\sum_t \log q_\psi(\mathrm{sg}(\boldsymbol{z}_t) \mid \mathrm{sg}(\boldsymbol{z}_{<t}))$
  Update $\phi, \boldsymbol{C}, \xi$ using $\mathcal{L}_{\mathrm{rec}} + \lambda \mathcal{L}_{\mathrm{align}}$
  Update ψ using $\mathcal{L}_\psi$
end for

Stage II: Train the generation prior.

Freeze $E_\phi, \boldsymbol{C}, D_\xi$; initialize a separate AR model p_θ.
$\mathcal{L}_{\mathrm{AR}}(\theta) \leftarrow -\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\frac{1}{n}\sum_{t=1}^{n} \log p_\theta(z_t \mid \boldsymbol{z}_{<t})\right]$
Train θ by minimizing $\mathcal{L}_{\mathrm{AR}}(\theta)$ on the frozen tokenizer's tokens.

4.2 Distributional Prior Divergence

TVC suggests a different formulation: global prior consistency requires only that the *marginal* distributions align. We retain the likelihood constraint (reconstruction) and relax the prior-matching term from a conditional to a distributional prior divergence (DPD):

$$\mathcal{L}_{DPD} = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\xi(\boldsymbol{x} \mid \boldsymbol{z})\right]\right] + D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z}))\,. \tag{2}$$

This retains reconstruction and adds aggregate prior matching during tokenizer training, following the distribution-level view used in adversarial and Wasserstein autoencoders (Tolstikhin et al., 2017) and information-regularized VAEs (Zhao et al., 2017).

Here p_θ is a randomly initialized, frozen 4-layer lightweight AR Transformer that provides a fixed, prefix-dependent target. The trainable decoder is $p_\xi(\boldsymbol{x} \mid \boldsymbol{z})$. The derivation uses the sequence-level KL; training averages the prior-matching term over tokens with $\lambda = 1.0$. Decoder gradients are obtained by standard backpropagation. The difficulty is obtaining a prior-matching gradient for hard tokens when the aggregate distribution $q_\phi(\boldsymbol{z})$ is unknown. Continuous-space matching methods also face gradient-estimation issues in this setting, discussed in Appendix C.1.

4.3 Effective Optimization of DPD

WGF-DPD applies Wasserstein Gradient Flow to the prior-matching functional $\mathcal{F}[q] = D_{\mathrm{KL}}(q\|p_\theta)$ (Jordan et al., 1998; Otto, 2001; Ambrosio et al., 2005; Santambrogio, 2015). The reconstruction term uses standard backpropagation.

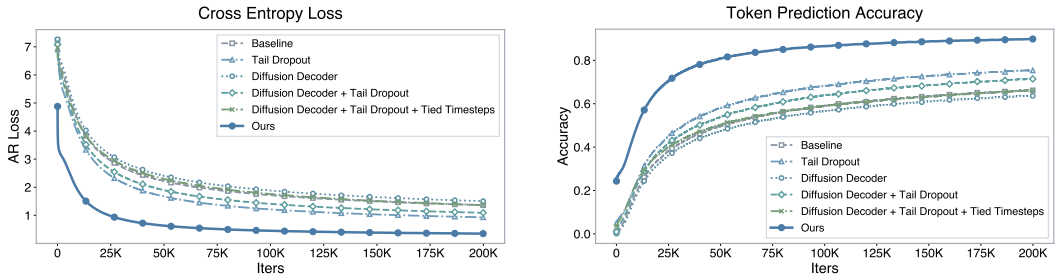
Using the chain rule through latent samples, we approximate the encoder gradient of the prior-matching term as

$$\nabla_\phi \mathcal{F}[q_\phi] \approx \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\left(\frac{\partial \boldsymbol{z}}{\partial \phi}\right)^{\top} \boldsymbol{g}(\boldsymbol{z})\right],$$

where $\boldsymbol{g}(\boldsymbol{z})$ is the prior-matching particle gradient. In the Wasserstein-2 geometry, the descent velocity is $\boldsymbol{v}(\boldsymbol{z}) = -\boldsymbol{g}(\boldsymbol{z})$, with

$$\boldsymbol{g}(\boldsymbol{z}) \triangleq \nabla_{\boldsymbol{z}}\left(\frac{\delta \mathcal{F}}{\delta q}\right)(\boldsymbol{z}).$$

The objective uses KL; Wasserstein-2 defines the geometry of the distribution update. Background on Wasserstein Gradient Flow is in Appendix E.



(a) Training loss curve.

(b) Top-1 accuracy.

Figure 1: AR training dynamics under matched iteration counts on CIFAR-10. (a) Cross-entropy loss; (b) Teacher-forcing top-1 accuracy. wAR-Tok achieves lower AR loss and higher prediction accuracy over the plotted iterations.

4.4 Gradient Flow Derivation and Tractable Approximation

The gradient of $\mathcal{L}_{\text{DPD}}$ with respect to a latent sample $\boldsymbol{z}$ comprises two components. The reconstruction term acts as a standard potential field. The prior matching term is a functional of the marginal distribution $q(\boldsymbol{z})$ and is optimized via Wasserstein Gradient Flow.

Applying the first variation to the KL divergence term $\mathcal{F}[q] = D_{\text{KL}}(q\|p_\theta)$, we obtain the particle gradient

$$\nabla_{\boldsymbol{z}} \frac{\delta \mathcal{F}}{\delta q(\boldsymbol{z})} = \nabla_{\boldsymbol{z}} \left(\log \frac{q(\boldsymbol{z})}{p_\theta(\boldsymbol{z})} + 1 \right) = \nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p_\theta(\boldsymbol{z}).$$

Combining with the reconstruction gradient, the total particle gradient becomes

$$\nabla_{\boldsymbol{z}} \mathcal{L}_{\text{DPD}} = \underbrace{-\nabla_{\boldsymbol{z}} \log p_\xi(\boldsymbol{x} \mid \boldsymbol{z})}_{\text{Reconstruction Score}} + \underbrace{\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p_\theta(\boldsymbol{z})}_{\text{Prior Matching Score}}. \tag{3}$$

The exact aggregate score $\nabla_{\boldsymbol{z}} \log q_\phi(\boldsymbol{z})$ is intractable: computing it requires integrating the conditional encoder over the entire data distribution, $q_\phi(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[q_\phi(\boldsymbol{z} \mid \boldsymbol{x})]$, and batch-level estimates would introduce high variance and bias.

To make the functional gradient computable, we introduce a parametric auxiliary model $q_\psi(\boldsymbol{z})$ to approximate the current aggregate posterior $q_\phi(\boldsymbol{z})$. Substituting this proxy into the gradient equation yields a tractable update:

$$\nabla_{\boldsymbol{z}} \mathcal{L}_{\text{DPD}} \approx \underbrace{-\nabla_{\boldsymbol{z}} \log p_\xi(\boldsymbol{x} \mid \boldsymbol{z})}_{\text{Reconstruction Score}} + \underbrace{\nabla_{\boldsymbol{z}} \log q_\psi(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p_\theta(\boldsymbol{z})}_{\text{Prior Matching Score}}.$$

Under standard Gaussian assumptions for the likelihood, the first term corresponds to the gradient of the Mean Squared Error (MSE). The latter terms are simply the score functions of the aggregate posterior proxy q_ψ and the reference p_θ. Optimization therefore requires only access to the *scores* (gradients) of these distributions, not their normalizing constants.

4.5 Score Estimation in Discrete Latent Spaces

For a categorical token with K possible values, the probability of a one-hot vector can be written as

$$P(\boldsymbol{z} \mid \boldsymbol{p}) = \prod_{k=1}^{K} p_k^{z_k}, \quad \boldsymbol{z} = (\delta_{1i}, \ldots, \delta_{Ki})$$

where p_k is the normalized probability of each class. Holding $\boldsymbol{p}$ fixed, we extend the log-probability linearly to relaxed token coordinates, giving

$$\nabla_{\boldsymbol{z}} \log P(\boldsymbol{z} \mid \boldsymbol{p}) = \frac{\partial \sum_{k=1}^{K} z_k \log p_k}{\partial \boldsymbol{z}} = \log \boldsymbol{p}$$

Table 1: CIFAR-10, comparison under **comparable reconstruction quality (rFID** ≈ 5.0). With matched rFID, wAR-Tok attains superior generation quality (gFID) and AR loss.

Representative Method	Iters (↓)	Training Techniques				rFID	gFID (↓)	gLoss (↓)
		FM Dec.	Tail Drop.	Tied Steps	WGF-DPD			
TiTok (Yu et al., 2024)	50K	✗	✗	✗	✗	4.89	44.42	1.07
ImageFolder/One-D-Piece (Li et al., 2024; Miwa et al., 2025)	90K	✗	✓	✗	✗	4.94	17.60	0.78
FlowMo (Sargent et al., 2025)	290K	✓	✗	✗	✗	4.91	38.68	1.50
FlexTok (Bachmann et al., 2025)	350K	✓	✓	✗	✗	5.07	24.23	1.13
SelfTok (Wang et al., 2025)	500K	✓	✓	✓	✗	5.01	42.60	1.41
wAR-Tok (Ours)	100K	✗	✗	✗	✓	4.99	**14.50**	**0.42**

Applying this token-level derivative with the AR predictions held fixed, the particle-level gradient for a discrete latent code $\boldsymbol{z}$ becomes

$$\nabla_{\boldsymbol{z}} \mathcal{L}_{\text{DPD}} \approx -\nabla_{\boldsymbol{z}} \log p_\xi(\boldsymbol{x} \mid \boldsymbol{z}) + \log \boldsymbol{Q}_\psi - \log \boldsymbol{P}_\theta. \tag{4}$$

Here $\log \boldsymbol{P}_\theta$ and $\log \boldsymbol{Q}_\psi$ denote the *per-token* log-probability vectors (over K codebook entries) induced by the AR factorizations $p_\theta(\boldsymbol{z}) = \prod_{t=1}^n p_\theta(z_t \mid \boldsymbol{z}_{<t})$ and $q_\psi(\boldsymbol{z}) = \prod_{t=1}^n q_\psi(z_t \mid \boldsymbol{z}_{<t})$.

To propagate this prior-matching signal to the encoder parameters ϕ, we must traverse the non-differentiable quantization step $\boldsymbol{z} = F_{\text{quant}}(\boldsymbol{h})$, where $\boldsymbol{z} \in \{0,1\}^K$ is the one-hot representation of a discrete token and $\boldsymbol{h} \in \Delta^{K-1}$ is its associated probability vector on the simplex. We obtain $\boldsymbol{h} = \mathrm{softmax}(\boldsymbol{\ell}/\tau_{\mathrm{ST}})$ from quantizer logits $\boldsymbol{\ell}$, either naturally arising in Gumbel-based quantizers (Jang et al., 2017; Maddison et al., 2017) or computed from vector-quantization distances (e.g., $\ell_k = -\|\boldsymbol{u} - \boldsymbol{c}_k\|_2^2$ for a latent embedding $\boldsymbol{u}$ and codebook entries $\{\boldsymbol{c}_k\}_{k=1}^K$), and define the hard assignment by $\boldsymbol{z} = \mathrm{onehot}(\arg\max_k h_k)$. Probability-STE (Bengio et al., 2013) treats $\partial \boldsymbol{z}/\partial \boldsymbol{h} \approx \boldsymbol{I}$ for the WGF term. Reconstruction uses the standard VQ straight-through gradient $\boldsymbol{g}_\phi^{\text{rec}}$. The mean-reduced encoder update is

$$\boldsymbol{g}_\phi = \boldsymbol{g}_\phi^{\text{rec}} + \lambda\, \mathbb{E}\left[\frac{1}{n}\sum_{t=1}^n \left(\frac{\partial \boldsymbol{h}_t}{\partial \phi}\right)^\top \left(\log \boldsymbol{Q}_{\psi,t} - \log \boldsymbol{P}_{\theta,t}\right)\right].$$

Computing the WGF term uses the two AR models' forward log-probabilities, without differentiating through their predictions or AR contexts. The proxy receives its own cross-entropy gradient on detached tokens in the same iteration.

4.6 Algorithm and Analysis

Algorithm 1 summarizes the two training stages. In Stage I, reconstruction and the detached log-probability contrast between q_ψ and the frozen p_θ update the tokenizer. The proxy q_ψ is trained on detached tokens with its own cross-entropy loss. In Stage II, we freeze the tokenizer and train a separate p_θ by AR cross-entropy. Here sg denotes stop-gradient and $\boldsymbol{C}\boldsymbol{z}$ denotes codebook embeddings. Token losses are averaged over positions and batch samples. Further details are in Appendix F.3.

Relation to distribution-matching distillation. In continuous space, the prior-matching gradient in (3) is the Wasserstein gradient of $D_{\mathrm{KL}}(q\|p)$. With diffusion-estimated scores, the same score-difference form appears in Distribution Matching Distillation (DMD) (Yin et al., 2024) and Diff-Instruct (Luo et al., 2023). Our discrete update uses the same score-difference structure through a token-level relaxation, giving the per-token log-probability vectors of (4).

Objective decomposition. The KL term combines reference likelihood with aggregate token entropy, which we estimate using the online proxy. The prior-matching term of objective 2 can be conceptually decomposed as

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\big[\log q_\psi(\boldsymbol{z}) - \log p_\theta(\boldsymbol{z})\big] = D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}) \,\|\, p_\theta(\boldsymbol{z})) - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}) \,\|\, q_\psi(\boldsymbol{z}))\,.$$

When $q_\psi = q_\phi$, the second KL term vanishes and the surrogate recovers the prior-matching objective. An imperfect proxy introduces estimation error, so we fit it to newly produced tokens at each step. Appendix F.3 gives the details.

Table 2: ImageNet 256×256 controlled comparison. Same encoder, decoder, and generation-prior architectures and training epochs; wAR-Tok adds WGF-DPD during tokenizer training. Lower is better for all metrics.

Tokenizer	rFID	gFID	Eval AR Loss
1D Tokenizer (baseline)	**2.05**	6.32	7.89
wAR-Tok (Ours)	2.35	**5.42**	**7.07**

5 EXPERIMENTS

We adopt 1D Tokenization as the foundational baseline, which provides a clean testbed without confounding spatial inductive biases. We compare *Tail Dropout*, *Diffusion Decoders*, and *Tied Timesteps* with *WGF-DPD* in a shared setup; the method names in the tables identify the corresponding techniques. We use a shared encoder, decoder backbone, and quantizer, with the input and timestep conditioning needed by diffusion decoders. Adversarial training is excluded from all baselines on CIFAR-10 to eliminate confounding instability. Full architectural details and the evaluation protocol (matched-rFID and matched-iteration) are in Appendix F. FID is computed with 50K samples following ADM (Dhariwal & Nichol, 2021).

5.1 CIFAR-10 RESULTS

Table 1 summarizes the matched-rFID protocol: wAR-Tok achieves the best gFID and the lowest AR generation loss (gLoss) at comparable reconstruction quality. The matched-iteration comparison (Appendix B, Table 3) also shows gains in generation quality and AR loss, so the improvement is not explained by longer tokenizer training. Figure 1 also shows lower AR loss and higher prediction accuracy for wAR-Tok during training. In this setup, tied timesteps require longer training to reach comparable rFID and yield higher gLoss than tail dropout alone.

Below wAR-Tok, tail dropout improves both gLoss and gFID, alleviating the AR hardness of the induced latents. Diffusion decoders, in contrast, converge substantially more slowly in Stage-I and do not improve downstream AR modeling in our setting: at similar rFID, diffusion-decoder variants exhibit higher gLoss than deterministic ones. While iterative diffusion sampling can partially compensate at inference and yield slightly better gFID than vanilla 1D tokenization, combining diffusion decoders with tail dropout makes the AR-hardness penalty more pronounced and yields worse gLoss/gFID than tail dropout alone.

5.2 IMAGENET 256×256 RESULTS

To check that the gains generalize beyond CIFAR-10, we run a controlled comparison on ImageNet 256×256. Both the baseline and wAR-Tok use a standard 1D-tokenizer architecture (encoder/decoder ∼86M parameters total) and a separate generation prior p_θ of ∼115M parameters trained for 300 epochs. wAR-Tok additionally uses a ∼40M-parameter online AR proxy during tokenizer training. Reconstruction losses are ℓ_1 (weight 1.0), LPIPS (weight 1.0), and PatchGAN (weight 0.1). Both models use AdamW with weight decay 3×10^{-2}. wAR-Tok adds the WGF-DPD update to tokenizer training, while retaining the reconstruction losses and the second-stage generation-prior training. Table 2 reports the results: rFID is slightly higher under wAR-Tok (2.35 vs. 2.05 for the baseline), gFID drops from 6.32 to 5.42 (about 14% relative improvement), and evaluation AR loss drops from 7.89 to 7.07. CFG schedules are tuned separately for each method; the settings are reported in Appendix F.2.

6 CONCLUSION

We use TVC to analyze the mismatch between reconstruction-trained tokenizers and AR priors. This view makes explicit that standard two-stage training leaves prior matching outside the tokenizer objective. We introduce WGF-DPD to add an aggregate prior-matching signal while keeping the reconstruction objective unchanged. For hard tokens, we approximate the update with a token-level

log-probability contrast between an online proxy and a frozen AR reference. The resulting tokenizer, wAR-Tok, reduces AR loss and improves generation quality on CIFAR-10 and ImageNet.

This view also helps explain existing alignment methods. In an idealized lossless bijection where $\boldsymbol{x} \equiv \boldsymbol{z}_{1:n}$, reconstructing from a prefix is equivalent to predicting the missing suffix: $\log p(\boldsymbol{x} \mid \boldsymbol{z}_{\le k}) = \sum_{j=k+1}^{n} \log p(z_j \mid \boldsymbol{z}_{<j})$. Tail dropout therefore resembles AR training in this special case. In practice, the target remains decoder reconstruction rather than prior likelihood, later tokens appear in more suffixes, and real tokenizers are not lossless bijections. SelfTok (Wang et al., 2025) ties prefix length to diffusion timestep and supervises tokens through a flow-matching decoder. Both methods shape tokens through reconstruction or denoising, while WGF-DPD introduces an explicit prior-matching signal.

WGF-DPD adds proxy-training cost and can trade reconstruction quality for easier AR modeling, so its weight requires tuning. The update relies on an approximate proxy and straight-through gradients. Integration with auxiliary tokenizer techniques, broader ImageNet comparisons, and extensions to other discrete priors remain future work. Our results show that including prior matching in tokenizer training can improve the reconstruction–generation trade-off.

AI USE STATEMENT

LLMs were used for language polishing, coding assistance, reviewing mathematical formulas, and searching for related literature. The core method and research were developed without AI assistance. The authors take full responsibility for the final content of this paper.

REFERENCES

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser, 2005.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL http://arxiv.org/abs/1701.07875v3.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length, 2025. URL http://arxiv.org/abs/2502.13967v2.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge–kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016. URL https://arxiv.org/abs/1511.06349.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer, 2022. URL https://arxiv.org/abs/2202.04200.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2105.05233.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models, 2019. URL http://arxiv.org/abs/1903.08689v6.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL https://arxiv.org/abs/2012.09841.

Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Wu. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.00887. URL https://arxiv.org/abs/1812.10907.

Tianyang Hu, Fei Chen, Haonan Wang, Jiawei Li, Wenjia Wang, Jiacheng Sun, and Zhenguo Li. Complexity matters: Rethinking the latent space for generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 29558–29579, 2023. URL https://arxiv.org/abs/2307.08283.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1611.01144.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359. URL https://doi.org/10.1137/S0036141096303359.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6114.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens, 2024. URL http://arxiv.org/abs/2410.01756v3.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2016. URL http://arxiv.org/abs/1608.04471v3.

James Lucas, G. Tucker, Roger Baker Grosse, and Mohammad Norouzi. Understanding posterior collapse in generative latent variable models. In *DGS@ICLR*, 2019. URL https://api.semanticscholar.org/CorpusID:156054774.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-Instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.18455.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1611.00712.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders, 2015. URL https://arxiv.org/abs/1511.05644.

Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression, 2025. URL http://arxiv.org/abs/2501.10064v1.

Felix Otto. The geometry of dissipative evolution equations: The porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174, 2001. doi: 10.1081/PDE-100002243. URL https://doi.org/10.1081/PDE-100002243.

Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2018. URL http://arxiv.org/abs/1803.00567v4.

Vivek Ramanujan, Kushal Tirumala, Armen Aghajanyan, Luke Zettlemoyer, and Ali Farhadi. When worse is better: Navigating the compression-generation tradeoff in visual tokenization, 2024. URL http://arxiv.org/abs/2412.16326v1.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. URL https://arxiv.org/abs/2102.12092.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015. URL https://arxiv.org/abs/1505.05770.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014. URL https://arxiv.org/abs/1401.4082.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.10752.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Birkhäuser, 2015.

Kyle Sargent, Kyle Hsu, Justin Johnson, Li Fei-Fei, and Jiajun Wu. Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19471–19481, 2025. URL https://arxiv.org/abs/2503.11056.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation, 2024. URL http://arxiv.org/abs/2406.06525v1.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2404.02905.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders, 2017. URL http://arxiv.org/abs/1711.01558v4.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018. URL http://arxiv.org/abs/1705.07120v5.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. URL https://arxiv.org/abs/1606.05328.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1711.00937.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.03762.

Bohan Wang, Zhongqi Yue, Fengda Zhang, Shuo Chen, Li'an Bi, Junzhe Zhang, Xue Song, Kennard Yanting Chan, Jiachun Pan, Weijia Wu, Mingze Zhou, Wang Lin, Kaihang Pan, Saining Zhang, Liyu Jia, Wentao Hu, Wei Zhao, and Hanwang Zhang. Selftok: Discrete visual tokens of autoregression, by diffusion, and for reasoning, 2025. URL http://arxiv.org/abs/2505.07538v3.

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior, 2024. URL http://arxiv.org/abs/2410.21264.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2024. URL https://arxiv.org/abs/2311.18828.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation, 2024. URL http://arxiv.org/abs/2406.07550v1.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec, 2021. URL http://arxiv.org/abs/2107.03312v1.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://arxiv.org/abs/1801.03924.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders, 2017. URL https://arxiv.org/abs/1706.02262.

A Impact Statement

Reducing the encoder–prior mismatch may lower the model size and cost needed for AR-based generation. We do not anticipate qualitatively new misuse risks beyond those already present in discrete generative models. Standard governance, content filtering, and ethical-oversight practices apply.

B Additional Experimental Results

Table 3 complements the main matched-rFID comparison by matching the number of tokenizer training iterations. Specifically, all methods are trained for the same number of iterations, which tests whether the gain of wAR-Tok comes from the proposed alignment objective rather than from longer optimization. Under this matched-iteration protocol, wAR-Tok keeps reconstruction quality comparable while substantially improving AR prior fit and generation quality.

Table 3: CIFAR-10, comparison under **identical tokenizer iteration counts (Iters = 400K)**. With the same iterations, wAR-Tok achieves comparable reconstruction (rFID) and superior generation (gFID) quality.

Representative Method	Iters	Training Techniques				rFID (↓)	gFID (↓)	gLoss (↓)
		FM Dec.	Tail Drop.	Tied Steps	WGF-DPD			
TiTok (Yu et al., 2024)	400K	✗	✗	✗	✗	2.30	52.53	1.38
ImageFolder/One-D-Piece (Li et al., 2024; Miwa et al., 2025)	400K	✗	✓	✗	✗	3.35	23.37	0.93
FlowMo (Sargent et al., 2025)	400K	✓	✗	✗	✗	3.41	38.63	1.52
FlexTok (Bachmann et al., 2025)	400K	✓	✓	✗	✗	4.42	22.76	1.09
SelfTok (Wang et al., 2025)	400K	✓	✓	✓	✗	6.49	44.79	1.37
wAR-Tok (Ours)	400K	✗	✗	✗	✓	3.06	**11.30**	**0.34**

Direct AR-loss training. We also jointly trained a tokenizer and an AR model with $\mathcal{L}_{\mathrm{rec}} + \lambda \mathcal{L}_{\mathrm{AR}}$, using a deterministic hard Gumbel-Softmax estimator. The 1D setup differs from that in Tables 1–3. In Table 4, $\lambda = 0.3$ improves generation with little change in reconstruction, while $\lambda = 3$ degrades both.

Table 4: Effect of the direct AR-loss weight. gFID is evaluated without CFG.

AR-loss weight λ	rFID (↓)	gFID (↓)
0	2.11	19.32
0.3	2.10	14.45
3	18.8	61.4

C Related Work

C.1 Comparison with Continuous-Space Distribution-Level Matching

Table 5 summarizes why existing distribution-level matching methods do not directly solve the setting considered in this work. These methods share our high-level goal of matching the aggregate posterior with a prior, but they are designed for continuous latent spaces and rely on gradients through kernels, discriminators, or reparameterized samples. In contrast, our setting uses hard discrete tokenizers and an autoregressive prior, where these gradient routes need a discrete relaxation or estimator. WGF-DPD provides a discrete-space alternative by using only forward AR log-probabilities to define the encoder update.

Table 5: Comparison with distribution-level aggregate-posterior matching methods. The table lists issues in directly applying continuous-space formulations to hard quantizers. "$\nabla_{\boldsymbol{z}} \log q$" denotes the analytic aggregate-posterior score.

	WAE-MMD (Tolstikhin et al., 2017)	**WAE-GAN/AAE** (Makhzani et al., 2015)	**InfoVAE** (Zhao et al., 2017)	**WGF-DPD (Ours)**
Latent space	Continuous	Continuous	Continuous	**Discrete**
Objective	MMD	Adversarial	Aggregate matching	KL
Gradient route	Backprop (kernel)	Backprop (discriminator)	Reparameterization	**Forward-pass via WGF**
Hard-token issue	Limited one-hot geometry	Discrete gradients; stability	Aggregate estimation	**Proxy + Prob-STE**

For one-hot tokens, common distance-based kernels do not distinguish different mismatched categories, and hard assignments block the usual pathwise gradient. Adversarial matching requires both a gradient estimator through the quantizer and stable discriminator training. InfoVAE-style aggregate matching also requires a tractable estimator for the chosen discrepancy. These issues motivate our proxy-based score contrast.

C.2 Further Related Work

Autoregressive modeling and large language models. AR modeling has been the dominant paradigm for large-scale sequence generation. Transformer architectures (Vaswani et al., 2017) enabled substantial scaling of next-token prediction. Extending AR to images dates back to PixelCNN (van den Oord et al., 2016). Recent systems push AR generation to text-to-image synthesis (Ramesh et al., 2021) and LLM-style backbones (Sun et al., 2024).

Discrete image generation. Discrete LVMs commonly follow a two-stage recipe: learn a tokenizer, then fit a prior over discrete codes (Esser et al., 2021). Masked token modeling provides an alternative to left-to-right decoding (Chang et al., 2022).

Wasserstein distance, gradient flow, and WAE. Optimal transport and Wasserstein distances provide geometry-aware discrepancies between distributions (Peyré & Cuturi, 2018). Wasserstein gradient flows formalize steepest descent over probability measures (Ambrosio et al., 2005). In generative modeling, Wasserstein objectives appear in WGAN (Arjovsky et al., 2017) and Wasserstein autoencoders (Tolstikhin et al., 2017), and motivate particle methods like SVGD (Liu & Wang, 2016).

Learnable priors and joint training. In LVMs, prior design crucially impacts representation learning and downstream generation. Standard variational objectives (Kingma & Welling, 2014) encourage posterior–prior matching but can suffer from posterior collapse (Bowman et al., 2016). This motivates learnable priors: VampPrior (Tomczak & Welling, 2018), normalizing-flow priors (Rezende & Mohamed, 2015), distribution-level aggregate-posterior matching (Tolstikhin et al., 2017), latent EBMs (Du & Mordatch, 2019), and divergence-triangle joint training (Han et al., 2019). Recent work emphasizes designing latents with downstream model complexity in mind (Hu et al., 2023).

1D tokenization and AR-alignment techniques. Beyond rasterized patches, several tokenizers use learnable holistic queries (Yu et al., 2024). Tail-dropout-style objectives (Bachmann et al., 2025) implicitly induce causal structure. Diffusion/flow-based decoders (Sargent et al., 2025) introduce sequential structure through iterative refinement. SelfTok (Wang et al., 2025) couples token prefix length with diffusion timesteps. LARP uses a trainable AR prior to guide tokenizer training (Wang et al., 2024), while CRT applies an ℓ_2 causal prediction loss to pre-quantization features (Ramanujan et al., 2024). Direct AR cross-entropy can decrease through lower token entropy as well as a closer fit to the token distribution.

D PROOFS

D.1 PROOF OF OTHER TWO CONDITIONS IN TVC

Case 1 (Likelihood + Prior ⇒ Posterior). The proof for Case 1 follows directly from the definition of the joint distribution.

Case 2 (Likelihood + Posterior ⇒ Prior). Assume $p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and $p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = q_\phi(\boldsymbol{x} \mid \boldsymbol{z})$. Then

$$\begin{aligned} p_\theta(\boldsymbol{z}) &= \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})} \\ &= \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) p_\theta(\boldsymbol{x})}{q_\phi(\boldsymbol{x} \mid \boldsymbol{z})} = \frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}\, q_\phi(\boldsymbol{z}). \end{aligned}$$

Assuming strict positivity on a common product support, the left-hand side does not depend on $\boldsymbol{x}$, so the ratio $p_\theta(\boldsymbol{x})/p_{\text{data}}(\boldsymbol{x})$ must be constant in $\boldsymbol{x}$. Integrating both sides over $\boldsymbol{z}$ yields $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$ and hence $p_\theta(\boldsymbol{z}) = q_\phi(\boldsymbol{z})$.

Case 3 (Posterior + Prior ⇒ Likelihood under completeness). Assume $p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and $p_\theta(\boldsymbol{z}) = q_\phi(\boldsymbol{z})$. Then

$$\begin{aligned} p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) &= \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{z})} = \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) p_\theta(\boldsymbol{x})}{q_\phi(\boldsymbol{z})} \\ &= q_\phi(\boldsymbol{x} \mid \boldsymbol{z})\, \frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})}. \end{aligned}$$

Integrating both sides over $\boldsymbol{x}$ gives

$$\mathbb{E}_{\boldsymbol{x} \sim q_\phi(\boldsymbol{x} \mid \boldsymbol{z})} \left[\frac{p_\theta(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})} \right] = 1, \quad \forall \boldsymbol{z}.$$

If the family $\{q_\phi(\boldsymbol{x} \mid \boldsymbol{z})\}_{\boldsymbol{z}}$ is complete, the above implies $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$ (a.e.), and substituting back yields $p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = q_\phi(\boldsymbol{x} \mid \boldsymbol{z})$.

D.2 Proof of ELBO as an Instance of TVC

Expanding the expected log-likelihood with TVC gives the ELBO and its variational gap:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\big[\log p_\theta(\boldsymbol{x})\big] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x}\mid\boldsymbol{z})}{q_\phi(\boldsymbol{x}\mid\boldsymbol{z})} + \log\frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z})} - \log\frac{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})}{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})} + \log p_{\text{data}}(\boldsymbol{x})\right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z}) - \log\frac{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})p_{\text{data}}(\boldsymbol{x})}{q_\phi(\boldsymbol{z})}\right. \\
&\qquad\qquad \left. + \log\frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z})} - \log\frac{p_\theta(\boldsymbol{z}\mid\boldsymbol{x})}{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})} + \log p_{\text{data}}(\boldsymbol{x})\right] \\
&= \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\Big[\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\big] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}\mid\boldsymbol{x})\,\|\,p_\theta(\boldsymbol{z}))\Big]}_{\text{ELBO}} \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\Big[D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}\mid\boldsymbol{x})\,\|\,p_\theta(\boldsymbol{z}\mid\boldsymbol{x}))\Big]}_{\text{Posterior Consistency Gap}}.
\end{aligned}
$$

This derivation demonstrates that the ELBO is fundamentally a specific instantiation of the TVC formalism, which targets Likelihood Consistency and Prior Consistency via KL divergence. The residual term corresponds to the well-known variational gap, which vanishes when the bound is tight. Consistent with our Constraint Redundancy analysis, this perspective confirms that explicitly satisfying likelihood and prior consistency provides a sufficient guarantee for posterior consistency.

D.3 Proof of ELBO Variants as an Instance of TVC

By the redundancy property (Corollary 3.2, Case 1), global optimality $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$ can be achieved by enforcing Likelihood Consistency and Prior Consistency. The standard ELBO enforces prior consistency through an instance-wise KL term $\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}\mid\boldsymbol{x})\|p_\theta(\boldsymbol{z}))$, which is sufficient but can be overly restrictive and induce posterior collapse.

A broad class of "ELBO variants" can be interpreted as replacing this instance-wise prior matching by a *distribution-level* penalty between marginals, optionally coupled with information/structure regularizers on the aggregate posterior:

$$
\max_{\theta,\phi}\ \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\big]\ -\ \lambda\,\mathcal{D}(q_\phi(\boldsymbol{z}),\,p_\theta(\boldsymbol{z}))\ +\ \gamma\,I_{q_\phi}(\boldsymbol{X};\boldsymbol{Z})\ -\ \eta\,\mathcal{R}(q_\phi(\boldsymbol{z}))\,,
$$

where $q_\phi(\boldsymbol{z}) \triangleq \mathbb{E}_{p_{\text{data}}}[q_\phi(\boldsymbol{z}\mid\boldsymbol{x})]$, $\mathcal{D}$ is a divergence between *marginals* (e.g., adversarial f-divergences, MMD, or Wasserstein), $I_{q_\phi}(\boldsymbol{X};\boldsymbol{Z})$ discourages the degenerate solution $\boldsymbol{z}\perp\boldsymbol{x}$, and $\mathcal{R}$ optionally regularizes aggregate structure (e.g., total correlation). Whenever optimizing such an objective yields (approximately) Likelihood Consistency $p_\theta(\boldsymbol{x}\mid\boldsymbol{z}) \approx q_\phi(\boldsymbol{x}\mid\boldsymbol{z})$ and Prior Consistency $p_\theta(\boldsymbol{z}) \approx q_\phi(\boldsymbol{z})$, the model enters the Case 1 consistency regime and therefore drives the same marginal objective $p_\theta(\boldsymbol{x}) \approx p_{\text{data}}(\boldsymbol{x})$.

D.4 Proof of Two-Stage Training as an Instance of TVC

For a fixed tokenizer, decoder fitting targets Likelihood Consistency, while prior fitting targets aggregate Prior Consistency. If both conditions hold exactly, Case 1 of Corollary 3.2 gives Posterior Consistency and $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$.

D.5 Proof of TVC

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})} \\
&= \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})} + \log \frac{q_\phi(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{x}, \boldsymbol{z})} \\
&= \log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})} + \log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) p_{\text{data}}(\boldsymbol{x})}{q_\phi(\boldsymbol{x} \mid \boldsymbol{z}) q_\phi(\boldsymbol{z})} \\
&= \underbrace{\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})}{q_\phi(\boldsymbol{x} \mid \boldsymbol{z})}}_{\text{Likelihood Consistency}} + \underbrace{\log \frac{p_\theta(\boldsymbol{z})}{q_\phi(\boldsymbol{z})}}_{\text{Prior Consistency}} \\
&\quad - \underbrace{\log \frac{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}}_{\text{Posterior Consistency}} + \log p_{\text{data}}(\boldsymbol{x})
\end{aligned}
$$

D.6 TVC as a Design Space

A central practical benefit of TVC is that it decouples the choice of objective along three orthogonal axes. **(A1) Consistency-pair decoupling.** Most existing methods target Likelihood–Prior. Corollary 3.2 identifies two additional valid pairings. **(A2) Divergence decoupling.** TVC is divergence-agnostic. Each consistency term may use a different divergence (KL, MMD, Wasserstein, f-divergence, ...). **(A3) Optimization decoupling.** Different consistency terms admit different schemes (reparameterization, adversarial, score matching, Wasserstein gradient flow, ...).

WGF-DPD (§4) sits at axis A2 (KL on the prior) and axis A3 (Wasserstein gradient flow). The following construction illustrates another possible objective.

Remark D.1 (A novel objective from TVC). **(A1)** By Case 2, enforcing exact Likelihood and Posterior Consistency yields $p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x})$. **(A2)** Assign KL to the likelihood term and MMD to the posterior term independently: $\mathcal{L} = \mathbb{E}_{\boldsymbol{z} \sim q_\phi}\big[D_{\mathrm{KL}}(q_\phi(\boldsymbol{x} \mid \boldsymbol{z}) \| p_\theta(\boldsymbol{x} \mid \boldsymbol{z}))\big] + \lambda\, \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\big[\mathrm{MMD}^2(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}), p_\theta(\boldsymbol{z} \mid \boldsymbol{x}))\big]$. **(A3)** Replace the MMD gradient with a normalized-kernel *drifting* field $\boldsymbol{V}(\boldsymbol{z} \mid \boldsymbol{x})$ such that matching $q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ implies zero drift, and replace the KL likelihood term with the standard reconstruction surrogate. The result is $\mathcal{L}_{\text{new}} = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z} \sim q_\phi}[-\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] + \lambda\, \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z} \sim q_\phi}\big[\big\|\boldsymbol{z} - \mathrm{sg}[\boldsymbol{z} + \boldsymbol{V}(\boldsymbol{z} \mid \boldsymbol{x})]\big\|^2\big]$, which lies outside the standard ELBO and two-stage menu.

E Wasserstein Gradient Flow

Under the Wasserstein-2 distance (Peyré & Cuturi, 2018), we define

$$
W_2^2(Q, P) = \inf_{\gamma \in \Gamma(Q,P)} \int_{X \times X} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \, \mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{y}).
$$

Introduce the continuity equation (Otto, 2001):

$$
\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t \boldsymbol{v}_t) = 0.
$$

The Wasserstein-2 distance admits an equivalent dynamic (minimum kinetic energy) formulation (Benamou & Brenier, 2000):

$$
W_2^2(\rho_0, \rho_1) = \inf_{\rho_t, \boldsymbol{v}_t} \left\{ \int_0^1 \int_X \|\boldsymbol{v}_t(\boldsymbol{x})\|^2 \rho_t(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \right\}
$$

subject to $\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t \boldsymbol{v}_t) = 0$, $\rho_{t=0} = \rho_0$, and $\rho_{t=1} = \rho_1$.

In this setting, the time derivative of an energy functional $F[\rho]$ is

$$\begin{aligned}\frac{\mathrm{d}F[\rho_t]}{\mathrm{d}t} &= \int_X \frac{\delta F}{\delta \rho_t}(\boldsymbol{x})\, \frac{\partial \rho_t}{\partial t}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \\ &= \int_X \frac{\delta F}{\delta \rho_t}(\boldsymbol{x})\, (-\nabla \cdot (\rho_t(\boldsymbol{x}) \boldsymbol{v}_t(\boldsymbol{x})))\, \mathrm{d}\boldsymbol{x} \\ &= \int_X \left(\nabla \frac{\delta F}{\delta \rho_t}(\boldsymbol{x}) \right) \cdot \boldsymbol{v}_t(\boldsymbol{x})\, \rho_t(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.\end{aligned}$$

Meanwhile, the Wasserstein gradient $\boldsymbol{V}$ is defined via

$$\frac{\mathrm{d}F[\rho_t]}{\mathrm{d}t} = \langle \boldsymbol{V}, \boldsymbol{v}_t \rangle_{L^2(\rho_t)} = \int_X \boldsymbol{V}(\boldsymbol{x}) \cdot \boldsymbol{v}_t(\boldsymbol{x})\, \rho_t(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.$$

Solving for $\boldsymbol{V}$ gives

$$\boldsymbol{V}(\boldsymbol{x}) = \nabla \left(\frac{\delta F}{\delta \rho_t}(\boldsymbol{x}) \right).$$

The corresponding gradient-flow velocity is

$$\boldsymbol{v}_t(\boldsymbol{x}) = -\boldsymbol{V}(\boldsymbol{x}) = -\nabla \left(\frac{\delta F}{\delta \rho_t}(\boldsymbol{x}) \right).$$

F Implementation Details and Architectures

For Tables 1–3, all methods use standard L2-normalized VQ and its STE for reconstruction. Only the WGF alignment gradient uses Prob-STE. All losses use mean reduction.

F.1 CIFAR-10 Setup

Architecture. We adopt a modern Transformer backbone across all components, incorporating GEGLU, RMSNorm, and AdaRMSNorm. Table 6 summarizes the dimensions used on CIFAR-10 (Krizhevsky, 2009). Each image is encoded into 62 tokens with a codebook of 1,024 entries.

Table 6: Trainable component sizes in our CIFAR-10 experiments; the frozen reference is described in Appendix F.3.

Component	Layers	Dim	Heads	MLP Ratio	Params
Encoder	10	512	8	4	∼50M
Decoder	10	512	8	4	∼50M
AR proxy q_ψ (joint)	4	512	8	4	∼17M
Generation prior p_θ (Stage II)	13	512	8	4	∼50M

Table 1 compares checkpoints at similar rFID, while Table 3 compares all methods at 400K tokenizer iterations.

Training. For Stage-I training on CIFAR-10, we train encoder and decoder for 400K iterations at batch size 512. For SelfTok we extend Stage-I by an additional 100K iterations to reach comparable rFID. For deterministic decoders we use weighted ℓ_1 (0.1), ℓ_2 (1.0), and LPIPS (Zhang et al., 2018) (0.1). For diffusion decoders, following FlowMo (Sargent et al., 2025), we use a flow-matching loss and compute LPIPS on $\hat{\boldsymbol{x}}_0$ converted from $\boldsymbol{v}$-prediction. For AR training on CIFAR-10, we use a 50M-parameter AR prior trained for a fixed 400K iterations. We use L2-normalized vector quantization (VQ) as the default quantizer.

We implement tail dropout following SelfTok: sample $t \sim \mathcal{U}(0, 1)$ and map to a discrete cut-off index using a SelfTok-like scheduler that biases toward later token positions. For methods with tail dropout but without tied timesteps, we apply it from the start with probability 0.5. For tied-timestep methods the ratio is determined by diffusion timesteps and tail dropout is applied at every iteration. We keep

diffusion and deterministic decoders architecturally identical except for input format and timestep conditioning through AdaRMSNorm.

WGF-DPD-specific details. Reconstruction uses standard VQ STE, while the WGF term uses Prob-STE to update the tokenizer. Cross-entropy on detached tokens updates the proxy. The reference stays frozen. Details are in Appendix F.3.

Stability. We use standard AdamW without additional stabilization heuristics. The tokenizer and proxy receive separate loss gradients in each iteration. WGF-DPD training remained stable in the reported experiments.

Compute. CIFAR-10 experiments run on $8\times$ NVIDIA RTX 5880 GPUs. Tokenizer training (about 400K iters, batch size 512) takes approximately one day. AR training (400K iters, $\sim$50M parameters) takes approximately half a day.

F.2 ImageNet 256 Setup

Losses. ℓ_1 (weight 1.0), LPIPS (weight 1.0), PatchGAN (weight 0.1).

Model sizes. Encoder/decoder $\sim$86M total (tokenizer); Stage-II generation prior p_θ $\sim$115M; Stage-I AR proxy q_ψ $\sim$40M.

Optimizer. AdamW with weight decay 3×10^{-2} for both the tokenizer (including the joint-training AR proxy) and the Stage-II generation prior.

Training. 150 epochs (tokenizer), 300 epochs (AR).

Compute. The training cost is equivalent to about 30 hours for the tokenizer and 20 hours for the AR prior on $8\times$ NVIDIA A100 GPUs.

Evaluation. FID-50K on the ImageNet validation set. We search the CFG schedule per method. The baseline uses a cosine schedule with scale 20 and power 1.5, and wAR-Tok uses a cosine schedule with scale 24 and power 2.0. The baseline uses the standard 1D tokenizer. rFID is computed on the validation set, and gFID follows the ADM protocol.

F.3 Frozen Reference and Online Proxy

The proxy q_ψ is trained at each step by cross-entropy on the tokenizer's current tokens, with the tokens detached for this loss. It estimates the current aggregate posterior. We use a $\sim$17M-parameter proxy on CIFAR-10 and a $\sim$40M-parameter proxy on ImageNet. Since the token distribution changes during training, this estimate is approximate.

The reference p_θ is a randomly initialized 4-layer lightweight AR Transformer that remains frozen throughout tokenizer training. It uses the same architecture as the proxy. Its role is to supply a fixed, prefix-dependent AR target. A target fitted to the same tokens could adapt to the tokenizer and weaken the matching signal. We apply temperature 0.1 to the reference logits before softmax. The online proxy and the Stage-II AR prior use temperature 1.0. The causal architecture supplies the ordering bias; freezing preserves the reference as the tokenizer changes. This reference is distinct from the Stage-II generation prior listed in Table 6.

For the tokenizer update, the per-token AR log-probability vectors are detached. Gradients pass through the relaxed token probabilities to the encoder and quantizer, without passing through the AR context. Proxy training still requires its own backward pass. In the reported CIFAR-10 setup, these additions incur about 10% extra cost over reconstruction-only tokenizer training. After Stage I, the encoder, codebook, and decoder are frozen, and p_θ is trained by AR cross-entropy for generation; the reference and proxy are discarded.

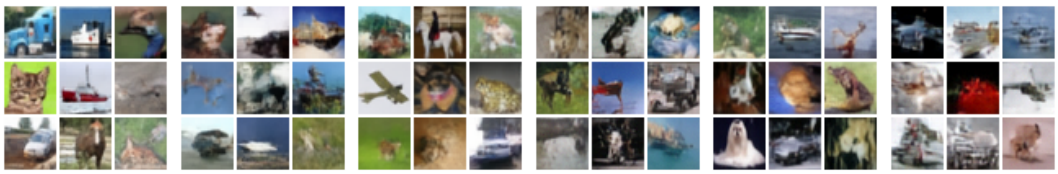
For Prob-STE, $\boldsymbol{h} = \mathrm{softmax}(\boldsymbol{\ell}/\tau_{\mathrm{ST}})$ with $\tau_{\mathrm{ST}} = 0.05$. The logits are negative squared distances between L2-normalized encoder features and codebook entries, with no additional logit scaling. We use $\lambda = 1.0$ and average alignment and AR cross-entropy losses over tokens and batch samples. The softmax Jacobian passes only the WGF gradient to the tokenizer logits; reconstruction follows the standard VQ STE path.

F.4 WHY WE FOCUS ON AR PRIORS HERE

TVC and DPD do not require an AR prior. Here, we derive and evaluate WGF-DPD using AR categorical predictions to study left-to-right token modeling. Extending the update to next-scale models such as VAR (Tian et al., 2024) would require matching their scale-wise factorization. For reparameterizable continuous latents, pathwise derivatives can replace STE, but estimating the aggregate and reference scores is still required.

G QUALITATIVE SAMPLES

We provide qualitative results in Figure 2. For a fair comparison, we use fixed random seeds across methods to control for sampling variability.



Ours gFID=11.30 | Baseline gFID=52.53 | Tail. gFID=23.37 | Diff. gFID=38.63 | Diff.+Tail. gFID=22.76 | Diff.+Tail.+Tied. gFID=44.79

Figure 2: Qualitative comparison on CIFAR-10. wAR-Tok produces samples with substantially better visual quality and the lowest gFID. “Diff.” = Diffusion Decoder; “Tail.” = Tail Dropout; “Tied.” = Tied Timesteps.