Guided End-to-End AutoRegression for Image Synthesis

Bin Lin[1,2,*] **Zheyuan Liu**[1,2,*] **Chenguo Lin**[1] **Sixiang Chen**[2,*] **Yunyang Ge**[1,2,*]
Yunlong Lin **Jianwei Zhang**[2] **Miles Yang**[2] **Zhao Zhong**[2] **Liefeng Bo**[2] **Li Yuan**[1,†]

[*]Work done during internship at Tencent Hunyuan [†]Corresponding author

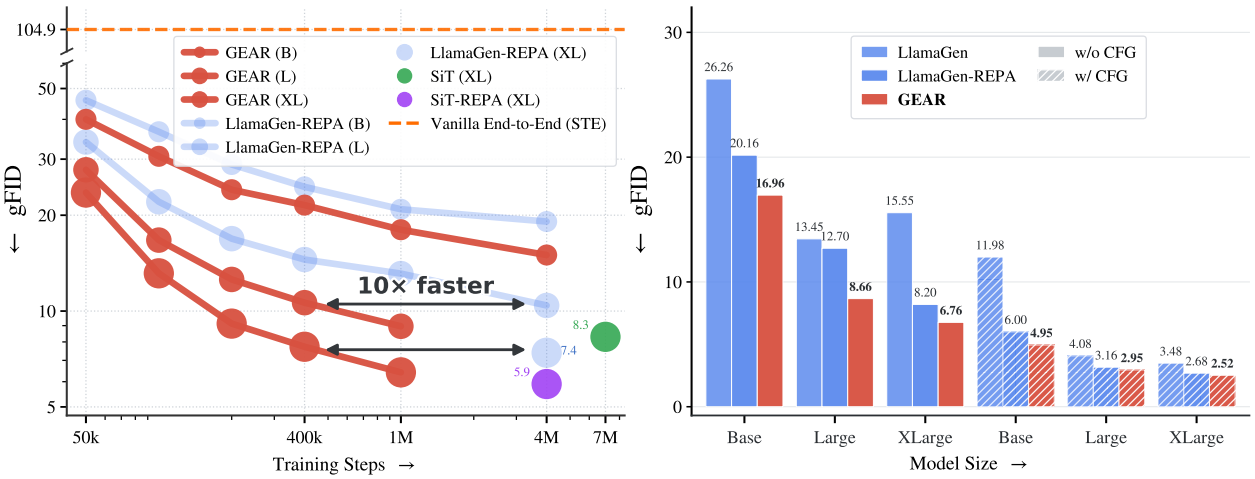
[1]Peking University, [2]Tencent Hunyuan

Abstract

Visual generative models are typically trained in two stages. A tokenizer is first trained for reconstruction and then frozen, after which a generator is trained on its discrete indices or continuous latents. This decoupling leaves the tokenizer unaware of what the generator finds easy to model. We present **GEAR** (**G**uided **E**nd-to-end **A**uto**R**egression), which trains a vector-quantized (VQ) tokenizer and an autoregressive (AR) generator jointly and end-to-end, guided by representation alignment. The key obstacle is that the VQ index fed to the AR model is non-differentiable, so gradients cannot reach the tokenizer, and a straight-through estimator collapses. GEAR resolves this with a dual read-out of the codebook assignment. A hard, one-hot branch trains the AR with next-token prediction, while a differentiable soft branch carries a representation-alignment loss that flows back to guide only the tokenizer. The AR model thereby steers its tokenizer toward an index distribution it can predict more easily. This shifts the alignment burden from the tokenizer to the AR: the tokenizer's own features become *less* DINOv2-like while the AR's become more so, the opposite of diffusion-side recipes that make the latent itself semantic. GEAR speeds up ImageNet gFID convergence by up to 10× relative to the strong LlamaGen-REPA baseline, learns markedly better patch-level and spatially-coherent features, and generalizes across quantizers (VQVAE, LFQ, IBQ) and to text-to-image generation.

GitHub Repo: https://github.com/Tencent-Hunyuan/GEAR

HuggingFace Model: https://huggingface.co/collections/BinLin203



(a) gFID vs. training steps (w/o CFG)

(b) gFID across model sizes (w/o vs. w/ CFG)

Figure 1 GEAR accelerates and improves autoregressive image generation. (a) gFID versus training steps on ImageNet (without CFG): GEAR converges up to 10× faster than LlamaGen-REPA, whereas the naive end-to-end variant that back-propagates into the tokenizer through the straight-through estimator diverges (gFID≈105). **(b)** gFID across model scales at 1.5M steps: GEAR improves performance at every size (B/L/XL), both without and with CFG.

1 Introduction

Modern visual generative models are almost universally trained in two stages. This holds both for autoregressive (AR) transformers over discrete tokens [40, 42] and for diffusion models over continuous latents [26, 31]. A tokenizer, either a VQ-VAE [44] or a continuous VAE, is first trained to reconstruct images. It is then *frozen*, and a generator is trained on the resulting indices or latents. The tokenizer is therefore optimized purely for reconstruction, oblivious to whether the latent space it induces is easy for the downstream generator to model.

This separation is convenient but suboptimal. The latent distribution is fixed by a reconstruction objective that does not know whether the induced sequence is easy to generate. The two goals are also in tension. Faithful reconstruction favors high-variance, detail-rich latents, whereas generation favors simple, predictable structure. Recent work begins to dissolve this boundary in the diffusion setting. A line of *representation alignment* methods accelerates training by injecting external semantics. REPA [48] aligns a diffusion model's intermediate features with a pretrained encoder such as DINOv2 [29], while VA-VAE and MAETok [4, 46] align the VAE *latent* itself with such features. Going further, REPA-E [20] trains the VAE and the diffusion model jointly end-to-end. Crucially, it shows that naively back-propagating the diffusion loss into the VAE *fails*, because the denoising objective flattens the latent variance that reconstruction needs. REPA-E therefore tunes the VAE end-to-end through the alignment loss rather than the diffusion loss.

The same opportunity exists for AR generation over discrete VQ tokens, but it is fundamentally harder. The map from a VQ index to the AR input is a non-differentiable arg max, so one cannot back-propagate any signal from the AR generator to the tokenizer, and the obvious remedy, a straight-through estimator (STE), is unstable and collapses the codebook in our joint setting (gFID≈105, figure 1). We propose **GEAR** (**G**uided **E**nd-to-end **A**uto**R**egression), which trains the VQ tokenizer and the AR generator jointly and end-to-end and resolves this with a dual read-out of the per-position codebook assignment. A *hard*, one-hot read-out reproduces the discrete tokens used at inference and trains the AR with next-token prediction and an alignment loss. A *soft* read-out is a temperature-weighted interpolation over the nearest codewords, and is therefore differentiable, carrying an alignment loss that flows back to update *only* the tokenizer. We never route the NTP loss into the tokenizer, because letting it reshape the codebook invites a collapse to a few low-entropy codes that trades reconstruction for predictability, as the concurrent EOSTok [6] also reports. The differentiable soft read-out is what makes any end-to-end gradient possible in this discrete setting, succeeding precisely where the STE collapses.

What this guidance actually does is, perhaps surprisingly, the opposite of the diffusion-side recipe. On the diffusion side, REPA-E, VA-VAE and MAETok make the *latent* more semantic by aligning it to a pretrained encoder. In GEAR the tokenizer's own features instead become *less* DINOv2-like, most strongly at the patch level (table 4). Rather than turning semantic, the tokenizer re-organizes its discrete index distribution toward a more predictable, lower-entropy usage (figure 4), without sacrificing reconstruction. The semantic alignment instead emerges inside the *AR generator*, whose hidden states track DINOv2 far more closely per patch and carry markedly more locally-coherent, spatially-causal structure, where LlamaGen-REPA [23] attains only global, image-level alignment (figure 5). End-to-end guidance thus shifts the alignment burden from the tokenizer to the AR: the tokenizer need not look semantic, only emit tokens the AR can predict, and this local, patch-level structure is exactly what makes next-token prediction easy, which explains GEAR's faster convergence and higher sample quality.

Our contributions are summarized as follows.

- **Guided end-to-end training.** We introduce GEAR, which jointly trains a VQ tokenizer and an AR generator. A differentiable soft-assignment bridge lets the AR generator's representation-alignment objective guide the tokenizer. This overcomes the non-differentiable index that defeats the straight-through estimator, and speeds up ImageNet gFID convergence by up to 10× relative to LlamaGen-REPA (figure 1).
- **Where representation alignment lives.** A representation analysis (section 4.3) shows that end-to-end guidance shifts the alignment burden from the tokenizer to the AR. Unlike diffusion-side recipes, GEAR's tokenizer becomes *less* DINOv2-like and re-organizes its index distribution toward predictable tokens, while the AR's features become more DINOv2-like *per patch*, with reconstruction preserved.

- **Generality.** The same mechanism works across quantizers, including VQVAE, LFQ [25, 47], and IBQ [36]. The end-to-end-tuned tokenizer is a drop-in that transfers across settings, and end-to-end training on ImageNet also accelerates text-to-image generation.

2 Related Work

Tokenizers and two-stage visual generation. Modern visual generators are built on a tokenizer that maps an image to a compact space, followed by a generative model over that space. Discrete pipelines pair a VQ tokenizer, such as VQ-VAE [44], VQGAN [10], LFQ [25, 47], or IBQ [36], with an autoregressive model [40, 42] or a masked generator [3]. Continuous pipelines pair a VAE with a diffusion transformer [26, 31]. In nearly all of these, the tokenizer is trained for reconstruction and then frozen, so the generator inherits a latent space it cannot influence. A few methods sidestep the discrete bottleneck by generating continuous tokens autoregressively [22], but the dominant, LLM-style route keeps discrete VQ tokens. We follow this discrete route and instead ask how the tokenizer itself should be shaped for it.

Representation alignment and semantic tokenizers. Semantics from self-supervised encoders such as DINOv2 [29], DINOv3 [38], SigLIPv2 [43], and V-JEPA2.1 [28] have become a powerful prior for generation. One family aligns the generator or the latent to such an encoder. REPA [48] aligns a diffusion model's intermediate features to DINOv2, while VA-VAE [46] and MAETok [4] align the VAE latent to these features. A second family makes the tokenizer itself semantic. RAE [52] pairs a frozen DINOv2 encoder with a trained decoder to obtain a representation latent for diffusion, VQRAE [9] discretizes such a representation autoencoder, and TA-Tok [16] and X-Omni [13] add vector quantization on top of a frozen SigLIP encoder. These semantic tokens speed up downstream generation and understanding, but because they are optimized for semantics rather than pixels, faithful reconstruction is hard, so TA-Tok and X-Omni rely on a separate generative de-tokenizer to render images. Recent analyses clarify which property actually matters. iREPA [39] finds that the *spatial structure* of the target representation, not its global semantic accuracy, drives generation, and PAE [50] finds that spatial-structure coherence and local continuity matter more than reconstruction fidelity. GEAR is consistent with these findings but reaches them differently. Instead of aligning to a fixed target offline, or trading away reconstruction by quantizing a semantic encoder, GEAR keeps a standard reconstruction tokenizer and lets the live AR model guide it end-to-end. This sharpens the *patch-level* structure of the AR's features (figure 5) and re-organizes the tokenizer's index distribution toward predictability (figure 4), without sacrificing reconstruction.

Toward end-to-end generative training. A growing line of work dissolves the two-stage boundary: REPA-E [20] jointly trains the VAE and the diffusion model, and pixel-space transformers drop the tokenizer to generate on raw pixels [8, 21, 24, 37, 49]. Seen this way, the dividing line is not latent versus pixel but whether the pipeline is trained end-to-end, echoing detection's shift from the multi-stage R-CNN [15] to single-stage, end-to-end detectors [1, 33]. The discrete VQ-AR setting is the hardest case, because the index is a non-differentiable arg max. Bridging it by sending the autoregressive gradient into the tokenizer through a straight-through estimator is unstable in our experiments (figure 1) and, as the concurrent EOSTok [6] also reports, collapses the codebook. The cause is a conflict the next-token-prediction (NTP) loss creates once it can update the tokenizer: NTP rewards a token sequence that is easy to predict, which the tokenizer can trivially obtain by driving its index distribution to low entropy and emitting a few dominant codes, whereas faithful reconstruction demands a high-entropy, fully-used codebook. When the prediction loss reaches the tokenizer it wins this trade-off, collapsing codebook utilization and reconstruction fidelity. The same shortcut appears in continuous form as the latent-variance collapse REPA-E reports for VAEs: in both cases a downstream generation objective, given access to the tokenizer, sacrifices fidelity for predictability. EOSTok mitigates this by down-weighting the next-token-prediction loss to a coefficient of 0.1 and adding an auxiliary pixel-space loss on its decoded AR predictions, which tacitly confirms that an undamped prediction loss on the tokenizer is harmful. GEAR instead removes the conflict at its source. The prediction loss never reaches the tokenizer: next-token prediction updates only the AR, while the tokenizer is shaped only by reconstruction and a representation-alignment signal carried through a differentiable soft assignment. Its codebook therefore stays broadly used rather than collapsing to a few dominant codes, and the AR trains and samples on exactly the discrete tokens it will see at inference.

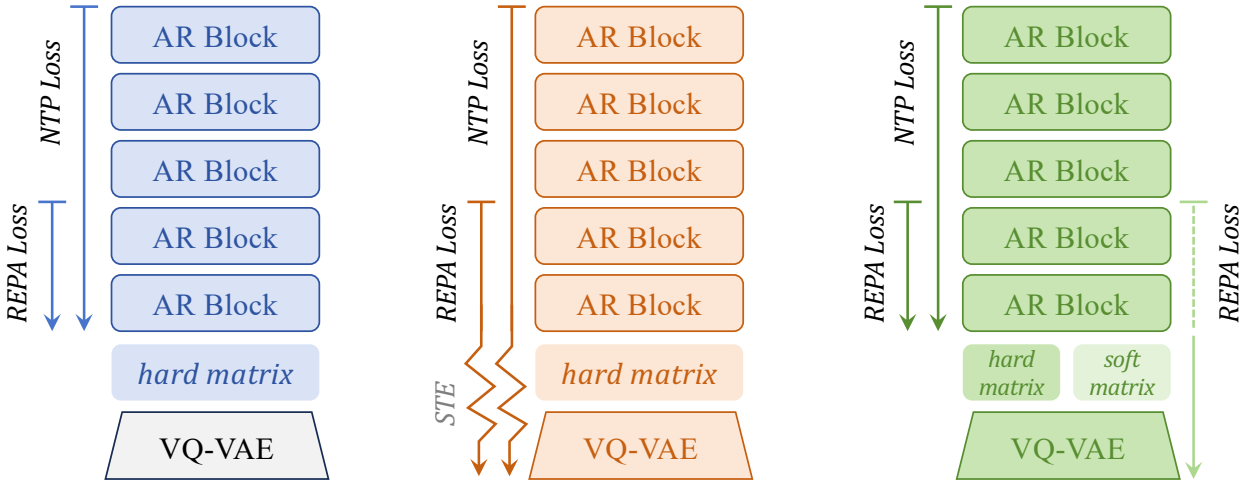


Figure 2 Overview of GEAR. (a) The conventional pipeline freezes a pretrained VQ-VAE and trains the AR model alone with the next-token-prediction (NTP) and REPA losses. **(b)** Naively making the pipeline end-to-end by passing AR gradients back into the tokenizer through the straight-through estimator (STE, drawn as the zigzag arrows ⇝) is highly unstable and collapses (cf. table 7). **(c)** GEAR reads the per-position assignment both as a *hard* (one-hot) and a *soft* (temperature-scaled) matrix: the hard branch carries NTP and the hard REPA loss to update only the AR model, while the differentiable soft branch carries a REPA loss that bypasses the upper AR blocks and flows back (the dashed arrow ⇢) to update only the tokenizer, giving a stable end-to-end guidance signal.

3 Method

We present **GEAR**, a framework that trains the visual tokenizer and the autoregressive (AR) generator *jointly and end-to-end*, so that the AR model can *guide* the tokenizer toward a discrete index distribution that is easier to model causally. We first review the two components that GEAR unifies (section 3.1), and then describe how a dual hard/soft assignment couples them through representation alignment while keeping their optimization cleanly decoupled (section 3.2). figure 2 contrasts GEAR with the conventional and naive end-to-end pipelines.

3.1 Preliminaries

Vector-quantized tokenization. A vector-quantized (VQ) tokenizer maps an image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ to a grid of discrete tokens. An encoder $\mathcal{E}$ produces a spatially down-sampled latent $\mathbf{Z} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{N\times d}$ with $N = h \times w$ positions (e.g., a 16× down-sampling factor yields $h = w = 16$ and $N = 256$). Each latent vector $\mathbf{z}_i$ is quantized to its nearest entry in a learnable codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^{K} \subset \mathbb{R}^d$,

$$q_i = \arg\min_k \|\mathbf{z}_i - \mathbf{c}_k\|_2^2, \qquad \hat{\mathbf{z}}_i = \mathbf{c}_{q_i}, \tag{1}$$

and a decoder $\mathcal{D}$ reconstructs $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{Z}})$. The tokenizer is trained with a reconstruction term, a perceptual (LPIPS [51]) term, an adversarial term [10], a codebook entropy term that encourages full codebook usage, and a commitment term:

$$\mathcal{L}_{\text{VQ}} = \mathcal{L}_{\text{rec}} + 0.1\,\mathcal{L}_{\text{LPIPS}} + 0.1\,\mathcal{L}_{\text{GAN}} + 0.05\,\mathcal{L}_{\text{ent}} + 0.25\,\mathcal{L}_{\text{commit}}. \tag{2}$$

Because the arg min assignment is non-differentiable, the gradient from $\mathcal{L}_{\text{rec}}$ to the encoder is classically approximated by the straight-through estimator (STE) [44].

Autoregressive image generation. Given the token grid $\mathbf{q} = (q_1, \ldots, q_N)$ produced by the tokenizer and flattened in raster order, together with a generic condition c (e.g., a class label or a text prompt), the AR

generator models the joint distribution over these *same* discrete indices causally,

$$p_\theta(\mathbf{q} \mid c) = \prod_{i=1}^{N} p_\theta(q_i \mid \mathbf{q}_{<i},\, c)\,. \tag{3}$$

The AR thus operates entirely at the level of discrete indices: its vocabulary is the set of K codebook entries, and it both conditions on and predicts the index q_i rather than the continuous code $\mathbf{c}_{q_i}$. For its input representation it does not reuse the VQ codebook $\mathcal{C}$. Instead, each index selects a row of the AR model's own learnable embedding table $\mathbf{E} \in \mathbb{R}^{K\times d}$, giving $\mathbf{u}_i = \mathbf{E}_{q_i}$. The condition is mapped to an embedding $\mathbf{e}_c$ and prepended, and the resulting sequence $\mathbf{S} = [\mathbf{e}_c, \mathbf{u}_1, \ldots, \mathbf{u}_N]$ is processed by a stack of L causal transformer blocks, producing hidden states $\mathbf{H}^{(\ell)} = (\mathbf{h}_1^{(\ell)}, \ldots, \mathbf{h}_N^{(\ell)})$ at every layer ℓ. A linear head predicts, over the same K indices, the distribution of the next token, trained by the next-token-prediction (NTP) loss

$$\mathcal{L}_{\text{NTP}} = -\sum_{i=1}^{N} \log p_\theta(q_i \mid \mathbf{q}_{<i}, c)\,. \tag{4}$$

At inference the model samples tokens autoregressively starting from c, and the decoder $\mathcal{D}$ renders the completed grid into an image.

Representation alignment. REPA [48] accelerates generative training by aligning an intermediate hidden state of the generator with features from a frozen, pretrained vision encoder f (e.g., DINOv2 [29]). Following its AR instantiation [23], a lightweight projection head g_ϕ maps the hidden state at a chosen depth ℓ into the target space, and the two are aligned token-wise by maximizing cosine similarity:

$$\mathcal{L}_{\text{align}}\left(\mathbf{H}^{(\ell)}\right) = -\frac{1}{N}\sum_{i=1}^{N} \frac{\left\langle g_\phi(\mathbf{h}_i^{(\ell)}),\, f(\mathbf{x})_i \right\rangle}{\|g_\phi(\mathbf{h}_i^{(\ell)})\|_2 \, \|f(\mathbf{x})_i\|_2}. \tag{5}$$

3.2 Guided End-to-End AutoRegression

Conventional pipelines train the tokenizer first and *freeze* it before training the AR generator. The discrete index distribution is therefore fixed by a pure reconstruction objective and is oblivious to whether the induced token sequence is easy to predict causally. GEAR removes this barrier: it optimizes the tokenizer and the AR model in a single end-to-end loop in which the AR model's representation objective *guides* the tokenizer. The central difficulty is that the index assignment is discrete, so gradients cannot flow from the AR model back to the encoder. We find that naively bridging this gap with the STE is highly unstable (table 7). We instead introduce a differentiable guidance channel built from a soft assignment.

Soft and hard assignments. For every position i the tokenizer produces an assignment vector $\mathbf{A}_i$ over the K codewords. Taking the simplest VQ tokenizer as the running example, we form it from the negative quantization distance to the codebook $\mathcal{C}$,

$$\mathbf{A}_{ik} = -\|\mathbf{z}_i - \mathbf{c}_k\|_2^2, \qquad \mathbf{A} \in \mathbb{R}^{N\times K}. \tag{6}$$

For tokenizers with a different quantization rule, such as LFQ [47] or IBQ [36], only the definition of $\mathbf{A}$ changes, and everything below is unchanged. We read $\mathbf{A}$ out in two complementary ways and map it onto the AR *embedding* table $\mathbf{E}$ (not the codebook $\mathcal{C}$): the *hard* read-out is the standard one-hot token lookup, whereas the *soft* read-out is a temperature-controlled mixture of embeddings,

$$\text{(hard)}\quad \mathbf{u}_i^{\text{h}} = \sum_k \mathbb{1}\left[k = \arg\max_{k'} \mathbf{A}_{ik'}\right] \mathbf{e}_k = \mathbf{E}_{q_i}, \tag{7}$$

$$\text{(soft)}\quad \mathbf{u}_i^{\text{s}} = \sum_k \boldsymbol{\pi}_{ik}\, \mathbf{e}_k = \boldsymbol{\pi}_i \mathbf{E}, \qquad \boldsymbol{\pi}_i = \text{softmax}(\mathbf{A}_i/\tau)\,, \tag{8}$$

where $\tau > 0$ is a guidance temperature and $\mathbf{e}_k$ is the k-th row of $\mathbf{E}$. Hence the discrete index is decided by the codebook $\mathcal{C}$, but the vector fed to the AR model is read from its own embedding table $\mathbf{E}$. The soft

Algorithm 1 One GEAR training step

Require: image $\mathbf{x}$, condition c, temperature τ, coefficient λ, depth ℓ

$\mathbf{Z} \leftarrow \mathcal{E}(\mathbf{x})$; $\quad \mathbf{A}_{ik} \leftarrow -\|\mathbf{z}_i - \mathbf{c}_k\|_2^2$ ▷ assignment to codebook $\mathcal{C}$

$\hat{\mathbf{z}} \leftarrow \mathbf{c}_{\arg\max \mathbf{A}}$; $\quad \hat{\mathbf{x}} \leftarrow \mathcal{D}(\hat{\mathbf{z}})$; $\quad$ evaluate $\mathcal{L}_{\mathrm{VQ}}$ ▷ VQ reconstruction

$\mathbf{u}^{\mathrm{h}} \leftarrow \mathbf{E}_{\arg\max \mathbf{A}}$; $\quad \mathbf{u}^{\mathrm{s}} \leftarrow \mathrm{softmax}(\mathbf{A}/\tau)\,\mathbf{E}$ ▷ AR hard / soft embeddings

forward $\mathbf{S}^{\mathrm{h}} = [\mathbf{e}_c, \mathbf{u}^{\mathrm{h}}]$ to layer L, and $\mathbf{S}^{\mathrm{s}} = [\mathbf{e}_c, \mathbf{u}^{\mathrm{s}}]$ to layer ℓ

evaluate $\mathcal{L}_{\mathrm{NTP}}$, $\mathcal{L}^{\mathrm{h}}_{\mathrm{align}}$ (hard branch) and $\mathcal{L}^{\mathrm{s}}_{\mathrm{align}}$ (soft branch)

$\theta_{\mathrm{tok}} \leftarrow \theta_{\mathrm{tok}} - \eta \nabla_{\theta_{\mathrm{tok}}} (\mathcal{L}_{\mathrm{VQ}} + \lambda \mathcal{L}^{\mathrm{s}}_{\mathrm{align}})$ ▷ guidance → tokenizer

$\theta_{\mathrm{AR}} \leftarrow \theta_{\mathrm{AR}} - \eta \nabla_{\theta_{\mathrm{AR}}} (\mathcal{L}_{\mathrm{NTP}} + \lambda \mathcal{L}^{\mathrm{h}}_{\mathrm{align}})$ ▷ NTP → AR

read-out $\mathbf{u}_i^{\mathrm{s}}$ is a fully differentiable function of the assignment $\boldsymbol{\pi}_i$ (hence of the encoder $\mathcal{E}$ and codebook $\mathcal{C}$) and of $\mathbf{E}$, and $\mathbf{u}_i^{\mathrm{s}} \to \mathbf{u}_i^{\mathrm{h}}$ as $\tau \to 0$. We use the hard read-out to define the discrete sequence the AR model must ultimately predict, and the soft read-out as a differentiable surrogate that carries gradients back into the original tokenizer.

Dual-branch forward. We build two condition-prefixed sequences,

$$\mathbf{S}^{\mathrm{h}} = \left[\mathbf{e}_c,\, \mathbf{u}_1^{\mathrm{h}}, \ldots, \mathbf{u}_N^{\mathrm{h}}\right], \qquad \mathbf{S}^{\mathrm{s}} = \left[\mathbf{e}_c,\, \mathbf{u}_1^{\mathrm{s}}, \ldots, \mathbf{u}_N^{\mathrm{s}}\right], \tag{9}$$

and feed both through the causal AR backbone. The hard branch passes through the non-differentiable arg max, so it carries no gradient to the tokenizer ($\mathcal{E}$, $\mathcal{C}$) and matches the discrete tokens used at inference. The soft branch instead keeps a live, differentiable connection to the tokenizer through $\boldsymbol{\pi}$. The hard branch is run to the final layer to produce next-token logits, whereas the soft branch only needs to reach the alignment depth ℓ and is *truncated* there, adding negligible compute.

Representation guidance. At depth ℓ both branches emit hidden states, $\mathbf{H}^{(\ell),\mathrm{h}}$ and $\mathbf{H}^{(\ell),\mathrm{s}}$, and we apply the alignment loss of equation (5) to each against the same target features $f(\mathbf{x})$:

$$\mathcal{L}^{\mathrm{h}}_{\mathrm{align}} = \mathcal{L}_{\mathrm{align}}\left(\mathbf{H}^{(\ell),\mathrm{h}}\right), \qquad \mathcal{L}^{\mathrm{s}}_{\mathrm{align}} = \mathcal{L}_{\mathrm{align}}\left(\mathbf{H}^{(\ell),\mathrm{s}}\right). \tag{10}$$

The two alignment terms play distinct roles. $\mathcal{L}^{\mathrm{h}}_{\mathrm{align}}$ regularizes the *generator* exactly as in REPA, operating on the hard, inference-time tokens. $\mathcal{L}^{\mathrm{s}}_{\mathrm{align}}$ is the *guidance signal*: because $\mathbf{u}^{\mathrm{s}}$ is differentiable, its gradient flows through the (otherwise fixed) AR backbone back to the encoder and the codebook, guiding the tokenizer to reshape its assignment so that the induced tokens are easier for the AR model to predict and to align with DINOv2.

Decoupled optimization. GEAR keeps the two modules' updates disjoint. The tokenizer parameters $\theta_{\mathrm{tok}} = \{\mathcal{E}, \mathcal{C}, \mathcal{D}\}$ are updated by the VQ objective (equation (2)) together with the soft guidance term, while the AR parameters $\theta_{\mathrm{AR}} = \{\text{transformer, embedding } \mathbf{E}, \text{condition embedding, head, } g_\phi\}$ are updated by NTP together with the hard alignment term:

$$\theta_{\mathrm{tok}} \;\leftarrow\; \theta_{\mathrm{tok}} - \eta\, \nabla_{\theta_{\mathrm{tok}}} \left(\mathcal{L}_{\mathrm{VQ}} + \lambda\, \mathcal{L}^{\mathrm{s}}_{\mathrm{align}}\right), \tag{11}$$

$$\theta_{\mathrm{AR}} \;\leftarrow\; \theta_{\mathrm{AR}} - \eta\, \nabla_{\theta_{\mathrm{AR}}} \left(\mathcal{L}_{\mathrm{NTP}} + \lambda\, \mathcal{L}^{\mathrm{h}}_{\mathrm{align}}\right), \tag{12}$$

with a single alignment coefficient λ. Concretely, the soft guidance gradient updates only the tokenizer (the AR backbone, embedding $\mathbf{E}$ and projector g_ϕ are held fixed for this term), while the non-differentiable arg max on the hard branch ensures that NTP and $\mathcal{L}^{\mathrm{h}}_{\mathrm{align}}$ update only the AR model. The encoder's *end-to-end* guidance therefore arrives through the differentiable soft assignment rather than the unstable STE, on top of the standard tokenizer objective $\mathcal{L}_{\mathrm{VQ}}$. equations (11) and (12) make the *guided* nature of GEAR explicit: a shared alignment target trains the generator on the hard, inference-time tokens while simultaneously steering the tokenizer through the soft, differentiable ones. We summarize the full details of procedure as shown in algorithm 1.

4 Experiment

4.1 Experimental Setup

Datasets and metrics. We study class-conditional image generation on ImageNet-1K at 256×256 resolution. For generation quality we report the generation FID (gFID), spatial FID (sFID), Inception Score (IS), Precision (Prec.) and Recall (Rec.), computed on 50K samples following the standard ADM evaluation protocol. To probe the co-trained tokenizer itself, we additionally report reconstruction FID (rFID), PSNR and SSIM. Unless stated otherwise, generation metrics are reported without classifier-free guidance (CFG). For text-to-image generation we first report a strictly controlled comparison on GPIC [2], where all methods are trained for a single epoch on the same 100M-image set. Following the official toolkit, on GPIC we report the Fréchet distance in DINOv2 feature space (FD-DINOv2, abbreviated FDD), Precision (Prec.), Recall (Rec.), Density (Dens.), Coverage (Cov.) and MMD. We additionally report the GenEval [14] and DPG-Bench [17] benchmarks, together with the CLIP Score and FID computed on the COCO 2017 validation set (5k image-text pairs). The CLIP Score is computed with the `openai/clip-vit-base-patch32` model.

Training protocol. We compare against LlamaGen [40], which couples a VQ tokenizer with a Llama-style causal transformer, and LlamaGen-REPA [23], which adds representation alignment to the AR model. Our LlamaGen-REPA and GEAR runs both start from the same warm-up tokenizer, a brief fine-tune that recovers the GAN discriminator omitted by public VQ tokenizers, following REPA-E [20], and barely changes reconstruction. From this shared start the two regimes differ. In our ablations and representation analysis, GEAR fine-tunes the tokenizer end-to-end jointly with the AR (equations (11) and (12)), whereas LlamaGen-REPA keeps it frozen. For the main results (tables 1 to 3) the long schedules make full joint training impractical, so GEAR instead freezes its end-to-end-tuned tokenizer, produced by the 400k-step joint run, and trains a fresh AR on top, exactly as LlamaGen-REPA trains on the frozen warm-up tokenizer. The two methods thus share an identical AR and training budget and differ only in the frozen tokenizer, so any gain isolates the tokenizer's contribution and shows that it transfers: the end-to-end-improved tokenizer can be dropped into a standard frozen-tokenizer pipeline without paying the end-to-end cost. To avoid confounding with model size, each scale uses its own matched tokenizer. Optimization and per-experiment hyperparameters are deferred to sections A and B.

Text-to-image instantiation. For text-to-image generation the condition is a sequence of text tokens from a Qwen3-1.7B [45] text encoder, following GPIC [2]. The model is a strict autoregressor over the concatenation of text and image tokens, trained from scratch on the 100M-image GPIC corpus. Its backbone, hybrid stream design and text-branch initialization are detailed in section B. GEAR and LlamaGen-REPA share this encoder and training and differ only in the frozen tokenizer. We report the GPIC toolkit metrics (table 2) and standard text-to-image benchmarks (table 3).

4.2 Main Results

Class-conditional ImageNet. table 1 compares GEAR with representative latent diffusion models (DiT [31], SiT [26], MDT [11, 12]) and autoregressive generators (LlamaGen, LlamaGen-REPA) on ImageNet 256×256, where the 111M/343M/775M rows of the AR models denote the B/L/XL variants. At a matched 300 epochs and parameter count, GEAR clearly improves over LlamaGen-REPA: with CFG, gFID drops from 6.00 to 4.95 (111M), 3.15 to 2.95 (343M) and 2.68 to 2.52 (775M), with consistently higher IS. End-to-end training also improves the tokenizer itself, with the per-scale behavior analyzed in the model-size ablation (table 9).

Text-to-image generation. We further apply GEAR to text-to-image synthesis, where the condition c in equation (4) is a text prompt. Our main, strictly controlled comparison is on GPIC [2] (table 2): every model uses the same Qwen3-1.7B [45] text encoder and is trained for a single epoch on the same 100M-image corpus, so differences reflect the method alone. Two observations stand out. First, at matched budgets GEAR consistently outperforms LlamaGen-REPA: across 50k/100k/200k/390k steps it lowers the GPIC FDD with CFG to 256.9/177.4/138.0/115.3 (vs. 279.6/198.6/153.5/127.9 for LlamaGen-REPA), mirroring the class-conditional gains. Second, autoregressive models converge far faster than the diffusion baseline: LlamaGen-REPA already surpasses the 390k-step JiT-GPIC result (FDD 204.0) after only 100k steps (FDD 198.6). We further report the same GEAR and LlamaGen-REPA models on the GenEval [14] and DPG-

Table 1 System-Level Comparison on Class-Conditional ImageNet 256 × 256. For the AR models, 111M/343M/775M denote the B/L/XL variants, and * marks inference at 384 resolution evaluated at 256. ImageNet val. is the real-data reference (an oracle floor). It is CFG-agnostic and shown once under the w/o-CFG columns. The w/ CFG results for LlamaGen-REPA and GEAR use a guidance scale of 1.5. The rows are grouped by (epochs, params), and within each group the best value across the three models is in **bold**.

Method	Epochs	Params	Generation w/o CFG					Generation w/ CFG				
			gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
ImageNet val. (ref.)	-	-	1.78	-	236.9	0.75	0.67	-	-	-	-	-
Latent Diffusion Models (LDM)												
DiT [31]	1400	675M	9.62	6.85	121.5	0.67	0.67	2.27	4.60	278.2	**0.83**	0.57
SiT [26]	1400	675M	8.61	6.32	131.7	0.68	0.67	2.06	4.50	270.3	0.82	0.59
MDT [11]	1300	675M	6.23	5.23	143.0	0.71	0.65	1.79	4.57	283.0	0.81	0.61
MDTv2 [12]	1080	675M	-	-	-	-	-	1.58	4.52	**314.7**	0.79	0.65
SiT-REPA [48]	800	675M	5.90	5.73	157.8	0.70	**0.69**	1.90	4.48	297.5	0.82	0.60
REPA-E [20]	800	675M	**1.69**	**4.17**	**219.3**	**0.77**	0.67	**1.12**	**4.09**	302.9	0.79	**0.66**
AutoRegressive (AR)												
LlamaGen [40]	300	111M	26.26	9.21	48.07	0.59	0.61	8.73	7.66	129.60	0.75	**0.53**
LlamaGen-REPA [23]	300	111M	20.16	7.03	60.66	0.64	**0.62**	6.00	5.77	144.98	0.79	**0.53**
GEAR (Ours)	300	111M	**16.96**	**5.55**	**67.62**	**0.67**	**0.62**	**4.95**	**5.08**	**166.13**	**0.82**	0.52
LlamaGen [40]	300	343M	13.45	8.32	82.28	0.65	0.63	4.07	8.15	198.50	0.80	0.55
LlamaGen-REPA [23]	300	343M	12.70	6.09	89.16	0.67	**0.65**	3.15	**5.26**	208.14	0.80	**0.57**
GEAR (Ours)	300	343M	**8.66**	**5.04**	**107.52**	**0.71**	0.63	**2.95**	5.40	**239.75**	**0.84**	0.54
LlamaGen [40]	300	775M	15.54*	7.04*	79.15*	0.61*	**0.68***	3.47*	5.81*	194.44*	0.76*	**0.60***
LlamaGen-REPA [23]	300	775M	8.20	5.56	115.06	0.70	0.65	2.68	5.45	232.15	0.81	0.58
GEAR (Ours)	300	775M	**6.76**	**4.96**	**129.00**	**0.72**	0.65	**2.52**	**5.14**	**262.94**	**0.84**	0.55
LlamaGen-REPA [23]	800	111M	19.14	7.52	64.27	0.64	**0.62**	5.30	5.83	158.87	0.79	**0.54**
GEAR (Ours)	800	111M	**14.98**	**5.27**	**73.63**	**0.68**	**0.62**	**4.35**	**4.96**	**180.85**	**0.82**	0.52
LlamaGen-REPA [23]	800	343M	10.44	5.85	100.54	0.68	**0.65**	2.92	5.40	215.99	0.80	**0.57**
GEAR (Ours)	800	343M	**8.61**	**5.07**	**109.80**	**0.71**	0.64	**2.72**	**5.22**	**240.30**	**0.83**	0.55
LlamaGen-REPA [23]	800	775M	7.46	5.74	124.49	0.71	**0.65**	2.57	5.36	236.27	0.81	**0.58**
GEAR (Ours)	800	775M	**6.28**	**4.99**	**136.73**	**0.73**	**0.65**	**2.45**	**5.05**	**254.27**	**0.83**	**0.58**

Bench [17] benchmarks, together with the CLIP Score and FID on the COCO 2017 validation set (table 3). In contrast to the distribution-level metrics (FDD, FID), GenEval, DPG-Bench and the CLIP Score probe *prompt adherence*, i.e. instruction following. The absolute scores are modest here for two reasons: the single-epoch model has not yet converged, and we train at 256 resolution but resize the samples to 512 for evaluation. As elsewhere, our aim is not to top these leaderboards but to compare methods under a controlled setting and isolate the effect of GEAR's end-to-end training. A full classifier-free guidance sweep on these benchmarks, with a discussion of this convergence behavior, is deferred to section D.

figure 3 shows where these GPIC gains come from. The tokenizer is frozen in both runs, so the two differ only in its quality. Even so, the AR trained on GEAR's tokenizer converges faster on both objectives: on GPIC it reaches the baseline's final REPA-alignment loss 11.1× faster and its NTP loss 2.5× faster. This acceleration is a property that the tokenizer carries over from end-to-end training: its tokens form a more predictable grid, so a freshly trained AR fits them with much less compute and reaches stronger patch-level DINOv2 alignment (figure 5).

Table 2 System-Level Comparison on Text-to-Image Generation (GPIC). All models share the same Qwen3-1.7B text encoder and are evaluated on the GPIC 50k test set with the official toolkit [2]. FDD is the Fréchet distance in DINOv2 feature space (FD-DINOv2). 390k steps corresponds to roughly one epoch over the 100M-image training set at a global batch size of 256. The w/ CFG columns use a guidance scale of 1.75. Rows are grouped by training steps, and within each group the better of LlamaGen-REPA and GEAR is in **bold**.

Method	Steps	Generation w/o CFG					Generation w/ CFG				
		FDD↓	Prec.↑	Rec.↑	Cov.↑	MMD↓	FDD↓	Prec.↑	Rec.↑	Cov.↑	MMD↓
JiT-GPIC-1.1B [2]	390k	-	-	-	-	-	204.0	0.91	0.53	0.80	-
LlamaGen-REPA-1.0B [23]	50k	414.8	**0.86**	0.33	0.52	0.59	279.6	**0.87**	0.46	0.67	0.35
GEAR-1.0B (Ours)	50k	**381.6**	**0.86**	**0.35**	**0.56**	**0.55**	**256.9**	**0.87**	**0.50**	**0.69**	**0.32**
LlamaGen-REPA-1.0B [23]	100k	319.2	0.86	0.44	0.63	0.44	198.6	0.89	0.61	0.77	0.23
GEAR-1.0B (Ours)	100k	**281.3**	**0.88**	**0.50**	**0.67**	**0.38**	**177.4**	**0.91**	**0.65**	**0.78**	**0.20**
LlamaGen-REPA-1.0B [23]	200k	261.4	0.88	0.56	0.68	0.34	153.5	0.90	0.72	0.82	0.17
GEAR-1.0B (Ours)	200k	**230.0**	**0.89**	**0.61**	**0.73**	**0.30**	**138.0**	**0.91**	**0.75**	**0.84**	**0.15**
LlamaGen-REPA-1.0B [23]	390k	228.9	0.89	0.63	0.74	0.30	127.9	**0.92**	0.78	0.85	0.14
GEAR-1.0B (Ours)	390k	**200.9**	**0.90**	**0.66**	**0.77**	**0.25**	**115.3**	**0.92**	**0.80**	**0.88**	**0.12**

Table 3 Text-to-Image Evaluation on Broader Benchmarks. The top block lists representative text-to-image systems for context. They differ in data, architecture, training schedule, inference (e.g., CFG scale), and tokenizer, and we report their published with-CFG GenEval (short prompts) and DPG-Bench. The bottom block is our controlled comparison: GEAR and LlamaGen-REPA (1.0B) share the same Qwen3-1.7B text encoder and GPIC training and differ only in the (end-to-end-tuned) frozen tokenizer. GenEval [14] is reported under *short* (original) and *long* (LLM-refined) prompts, and the CLIP Score and FID are on the COCO 2017 validation set (5k pairs). Within the controlled comparison the better of the two is in **bold**, and "–" marks numbers not reported.

Method	Generation w/o CFG					Generation w/ CFG				
	GenEval↑		DPG-Bench↑	CLIP Score↑	FID↓	GenEval↑		DPG-Bench↑	CLIP Score↑	FID↓
	short	long				short	long			
Other systems (different data, architecture, schedule, inference, and tokenizer)										
PixArt-α-0.6B [5]	-	-	-	-	-	0.48	-	71.11	-	-
Chameleon-7B [41]	-	-	-	-	-	0.39	-	-	-	-
Controlled comparison (same data, architecture, schedule, and inference, differing only in the tokenizer)										
LlamaGen-REPA-1.0B [23]	0.074	0.218	**55.386**	27.12	30.04	0.272	0.419	70.363	30.80	27.99
GEAR-1.0B (Ours)	**0.086**	**0.227**	55.369	**27.29**	**29.66**	**0.334**	**0.478**	**72.881**	**31.39**	**25.74**

4.3 Representation Analysis

To understand *why* guiding the tokenizer with the AR model helps, we probe the tokenizer's own DINOv2 feature similarity (table 4) and its codebook usage over training (figure 4), together with the DINOv2 similarity of the AR's per-layer hidden states (figure 5). Similarities are measured with CKNNA [18] and CKA [19]. The emerging picture is the opposite of the diffusion-side recipe: end-to-end guidance leaves the tokenizer *less* DINOv2-like, not more, and relocates the alignment into the AR.

The tokenizer becomes less DINOv2-like. table 4 shows that end-to-end training makes the tokenizer's features *less* similar to DINOv2, not more. GEAR is below the warm-up tokenizer at every entry, and the gap is far larger at the patch level (CKA 0.173 to 0.107) than at the image level. This is a re-organization rather than a loss of information, since reconstruction is preserved (table 8). It is also harmless for generation, because the AR never consumes these continuous features. The AR ingests only the discrete index sequence

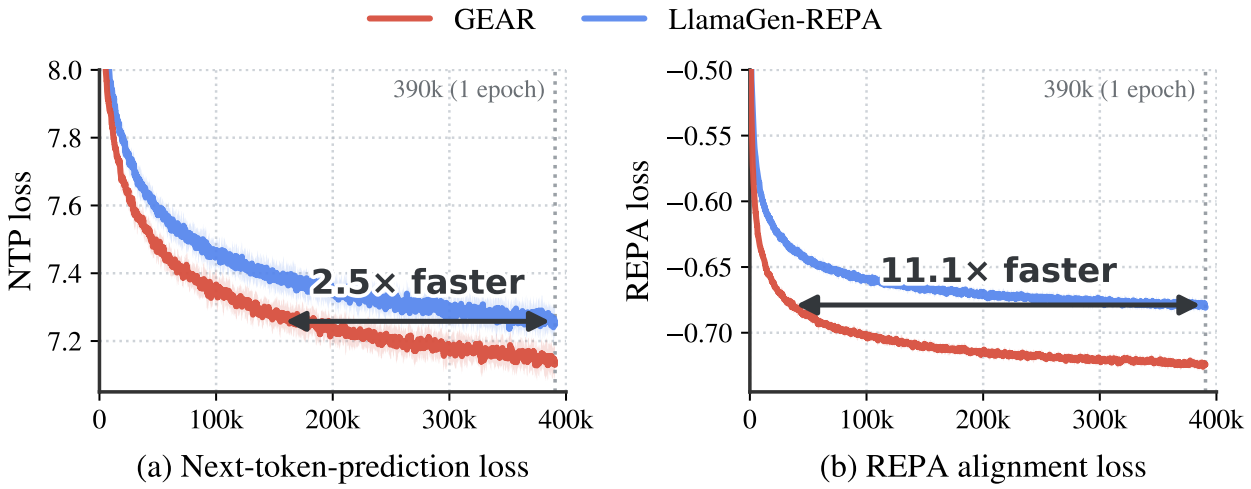


Figure 3 Training dynamics on GPIC. NTP loss (a) and REPA alignment loss (b) over the single epoch (390k steps). We train an AR on GEAR's end-to-end-improved tokenizer and on the original tokenizer. Both are kept frozen here, so the two runs differ only in tokenizer quality. To reach the baseline's final loss, GEAR is 2.5× faster on NTP and 11.1× faster on REPA alignment.

Table 4 Tokenizer-feature alignment to DINOv2. CKNNA and CKA between the tokenizer's own features and DINOv2 on the ImageNet validation set, before (pre) and after (post) quantization, at the image and patch level. Higher is more DINOv2-like. LlamaGen-REPA is the warm-up (non-end-to-end) tokenizer, and parenthesized values are GEAR's change relative to it.

Metric	Level	Pre-quantization		Post-quantization	
		LlamaGen-REPA	GEAR	LlamaGen-REPA	GEAR
CKNNA	image	0.2485	0.2440 (−0.0045)	0.2470	0.2410 (−0.0059)
	patch	0.1192	0.0831 (−0.0361)	0.1054	0.0748 (−0.0307)
CKA	image	0.1863	0.1805 (−0.0058)	0.1846	0.1784 (−0.0062)
	patch	0.1727	0.1070 (−0.0657)	0.1609	0.0997 (−0.0612)

through its own input embedding, so what it needs is not a DINOv2-like tokenizer but a discretization whose tokens it can predict and align with DINOv2. **Contrast this with latent diffusion.** REPA-E [20], VA-VAE [46] and MAETok [4] all improve diffusion by making the continuous VAE latent itself *more* semantic and more aligned to vision-foundation features. Because a discrete AR consumes indices rather than a continuous latent, GEAR benefits from the reverse move. It reshapes the tokenizer for token predictability and lets the semantic alignment emerge inside the AR.

The codebook usage sharpens. End-to-end training also reshapes *which* codes the tokenizer uses, in the direction of predictability. figure 4 tracks the codebook-assignment statistics over the joint fine-tuning run (τ=0.1), starting from the warm-up tokenizer (blue). The AR rapidly sharpens the usage distribution: the cumulative top-1/10 assignment mass rises while the usage entropy and the effective codebook size fall, peaking in concentration around 30k steps before relaxing and converging (after ∼120k) to a distribution that stays more concentrated and lower-entropy than the warm-up. A lower-entropy, more peaked token distribution is exactly an easier next-token target, so end-to-end training re-tunes the tokenizer to emit more predictable tokens. Because this pressure comes from the alignment signal rather than the prediction loss, it stops well short of the few-code collapse that next-token prediction would induce: the codebook stays broadly used and reconstruction is preserved (table 8).

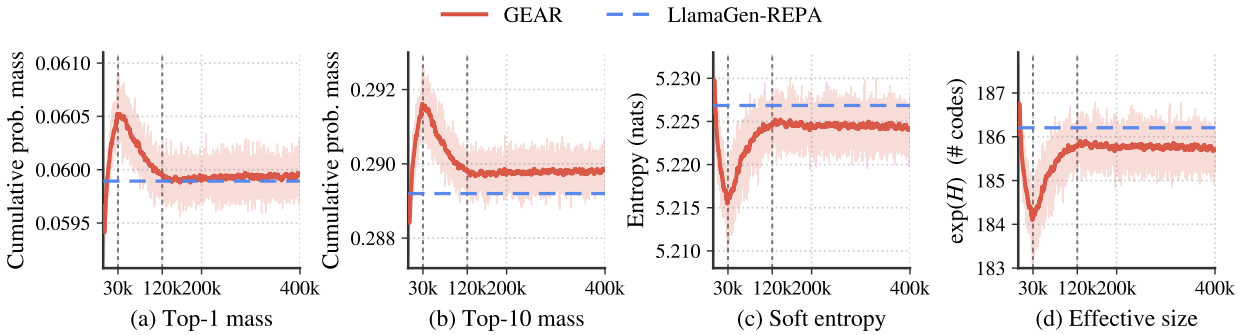


Figure 4 Codebook usage during end-to-end fine-tuning. Over GEAR's joint fine-tuning on ImageNet (τ=0.1), we track the cumulative top-1/10 assignment mass (a–c), the usage entropy in nats (d), and the effective codebook size $\exp(H)$ (e). The blue dashed line marks the warm-up tokenizer, the start of fine-tuning that LlamaGen-REPA uses. The distribution sharpens rapidly, peaks in concentration near 30k steps, then relaxes and converges (after ∼120k) to a state more concentrated and lower-entropy than the warm-up.

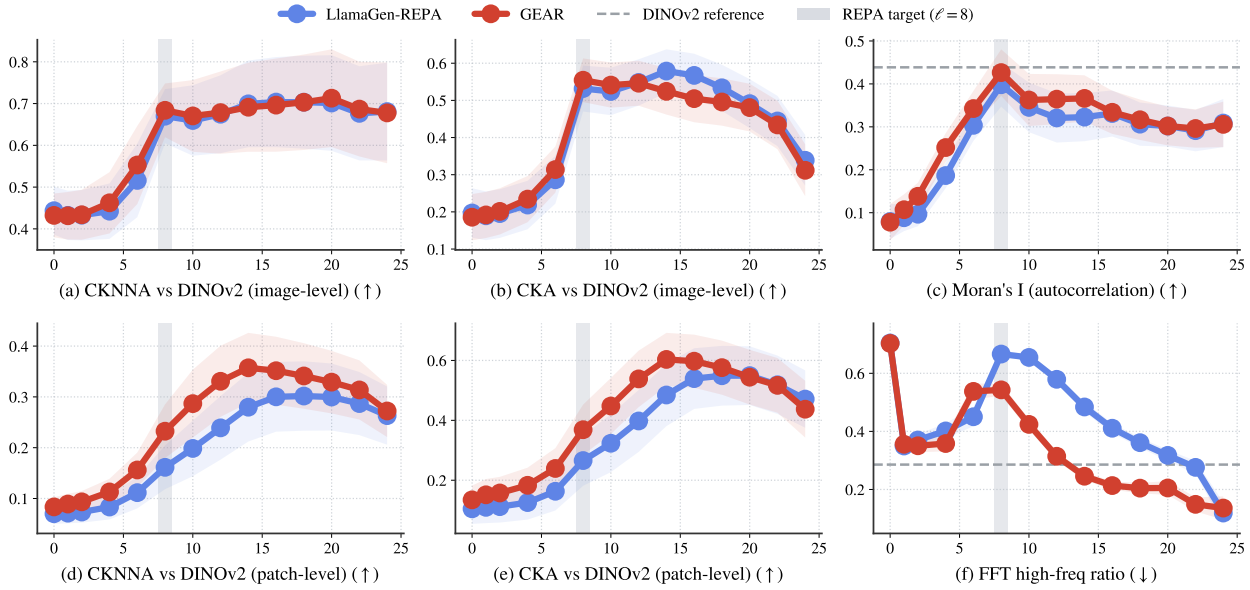


Figure 5 AR-feature alignment to DINOv2 across depth. GEAR-L vs. LlamaGen-REPA-L, both aligning the 8-th layer to DINOv2 (gray band) and trained for 400k steps. The horizontal axis is the AR layer index, where layer 0 is the output of the AR embedding layer. We take the *raw* per-layer hidden states (before the REPA projection head) and measure alignment to DINOv2 via CKNNA [18] and CKA [19] at the image level (a,b) and patch level (d,e), together with Moran's I [27] (c) and the FFT high-frequency ratio (f). The gray dashed line is DINOv2's own value, and the shaded bands denote ±1 standard deviation. The two models are comparable at the image level, but GEAR is markedly closer to DINOv2 at the patch level (d,e) and in both locality statistics (c,f), showing that end-to-end guidance yields features with stronger local, spatially-causal structure.

Image-level vs. patch-level alignment. The AR's hidden states tell the opposite story. At the image level (a,b), the CKNNA and CKA curves of GEAR and LlamaGen-REPA are nearly indistinguishable across depth, both peaking around the alignment layer: explicit alignment already makes the two models comparably DINOv2-like in their global, pooled semantics. The picture changes sharply at the patch level (d,e): GEAR (red) lies well above LlamaGen-REPA (blue) across the mid-to-deep layers, for *both* CKNNA and CKA. In other words, the two methods agree with DINOv2 about the image as a whole, but GEAR matches DINOv2 far more closely *per patch*, indicating that end-to-end guidance injects richer local structure into each token.

Table 5 Ablation Study on Guidance Temperature.

Temperature	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
0.500	**9.153**	5.225	**101.017**	**0.719**	0.620	20.571	0.552	1.698
0.100	10.630	5.111	92.700	0.701	0.630	20.779	0.558	1.640
0.050	10.745	**4.930**	93.024	0.696	0.626	20.804	0.558	1.638
0.010	11.424	4.991	90.201	0.698	0.628	20.825	0.559	1.628
0.005	11.413	5.072	89.801	0.697	**0.635**	20.825	0.559	1.628

Table 6 Ablation Study on Alignment Coefficient.

Coefficient	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
0.25	12.005	5.165	87.183	0.691	**0.637**	20.912	0.564	1.653
0.50	**10.630**	**5.111**	**92.700**	**0.701**	0.630	20.779	0.558	1.640
0.75	10.819	5.426	92.451	0.696	0.620	20.618	0.551	1.647
1.00	10.889	5.507	91.086	0.696	0.633	20.465	0.546	1.715

Locality and spatial autocorrelation. Panels (c) and (f) corroborate this from a signal-processing view. Both Moran's I (c) and the FFT high-frequency ratio (f) measure how spatially self-correlated, or locally causal, the feature map is. GEAR tracks the DINOv2 reference (gray dashed) much more closely than LlamaGen-REPA: its Moran's I stays near the DINOv2 level at depth, and its high-frequency energy descends toward DINOv2's ratio rather than remaining elevated and noisy as for LlamaGen-REPA. Because autoregressive generation predicts each token causally from its local spatial context, tokens whose features carry coherent, DINOv2-like local structure are inherently easier to predict. This is the empirical signature of the AR model successfully reshaping the tokenizer's index distribution (figure 4), and it explains the consistent FID gains reported above.

Where the alignment lives. Putting the two probes together, end-to-end training does not make the tokenizer more DINOv2-like. Instead it shifts the alignment burden from the tokenizer to the AR. The gradient routed back through the soft assignment reshapes the tokenizer toward a discretization the AR can predict and align, rather than toward DINOv2-like continuous features. As a result the tokenizer's own DINOv2 similarity drops (table 4) while the AR's rises, most clearly per patch (figure 5). This is why a frozen end-to-end tokenizer trains a faster-converging and higher-quality AR (figure 3 and table 1) even though, on its own, it looks *less* semantic. Reading down the depth axis, the global metrics (a,b,c) peak near the alignment layer while the patch-level curves (d,e) keep strengthening into the deeper layers, and GEAR stays at least as close to DINOv2 as LlamaGen-REPA at *every* depth. A detailed account of where each curve peaks, including the raw-feature high-frequency bump at ℓ=8 in (f), is given in section E.

4.4 Ablation Studies

Each ablation table also reports tokenizer reconstruction: the warm-up tokenizer used to recover the GAN discriminator (the LlamaGen-REPA row) and GEAR's end-to-end-tuned tokenizer (the GEAR row). Across all studies, end-to-end training preserves and often slightly improves reconstruction. section C details the official tokenizers and how the evaluation interpolation affects these reconstruction metrics.

Guidance temperature. table 5 sweeps the temperature τ of the soft assignment. As τ shrinks, the soft batch collapses onto the hard one and the guidance signal weakens, so gFID degrades monotonically below τ=0.1 while rFID improves only marginally. A large τ=0.5 yields the lowest gFID (9.153) but the loosest soft-hard consistency and the worst reconstruction (1.698 rFID). We adopt τ=0.1 as a robust default that balances generation and reconstruction across the other studies.

Alignment coefficient. table 6 varies the coefficient λ that weights both alignment terms. Performance is best at λ=0.5 on essentially every metric: a smaller λ=0.25 under-uses the guidance, while a larger λ=1.0 over-regularizes and hurts both generation and reconstruction (rFID 1.715). We therefore set λ=0.5.

Table 7 Ablation Study on Role of Different Components.

Component	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
w/ STE	104.932	31.933	12.418	0.292	0.233	12.596	0.239	59.723
w/o $\mathcal{L}_{\text{GAN}}$	16.353	8.084	76.346	0.649	0.590	22.343	0.606	5.857
GEAR (Ours)	**10.630**	**5.111**	**92.700**	**0.701**	**0.630**	20.779	0.558	1.640

Table 8 Ablation Study on VQ Tokenizers.

Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
VQVAE [40, 44]	14.719	6.009	77.333	0.670	**0.640**	21.061	0.565	1.724
+GEAR (Ours)	**10.630**	**5.111**	**92.700**	**0.701**	0.630	20.779	0.558	1.640
LFQ [25, 47]	18.681	6.525	65.265	0.666	**0.625**	20.969	0.562	2.421
+GEAR (Ours)	**14.776**	**5.553**	**74.512**	**0.690**	0.621	20.477	0.550	2.129
IBQ [36]	20.246	6.897	61.628	0.638	**0.634**	21.182	0.577	1.973
+GEAR (Ours)	**12.972**	**5.165**	**80.307**	**0.679**	0.629	20.917	0.567	1.716

Table 9 Ablation Study on Model Size.

Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
LlamaGen-REPA-B [23]	24.986	6.762	49.086	0.620	**0.614**	21.061	0.565	1.724
+GEAR (Ours)	**21.516**	**5.420**	**54.397**	**0.644**	0.610	20.751	0.556	1.658
LlamaGen-REPA-L [23]	14.719	6.009	77.333	0.670	**0.640**	21.061	0.565	1.724
+GEAR (Ours)	**10.630**	**5.111**	**92.700**	**0.701**	0.630	20.779	0.558	1.640
LlamaGen-REPA-XL [23]	9.631	5.556	104.456	0.697	**0.644**	21.061	0.565	1.724
+GEAR (Ours)	**7.693**	**4.934**	**115.463**	**0.717**	0.637	20.808	0.559	1.624

Role of different components. table 7 isolates two design choices. Replacing our differentiable soft-assignment bridge with the conventional straight-through estimator destabilizes the joint training catastrophically, collapsing both generation (gFID 104.9) and reconstruction (rFID 59.7). This confirms that the soft assignment is essential as the differentiable channel that carries AR gradients into the tokenizer. Removing the adversarial loss is far less severe but still markedly degrades both reconstruction (1.640→5.857 rFID) and generation (10.630→16.353 gFID), confirming that adversarial supervision remains important for the co-trained tokenizer.

VQ tokenizers. table 8 shows that GEAR is largely agnostic to the underlying quantizer, improving all three: on VQVAE [40, 44] gFID drops 14.72 → 10.63 (rFID 1.724 → 1.640), on LFQ [25, 47] 18.68 → 14.78 (rFID 2.421 → 2.129), and on IBQ [36] 20.25 → 12.97 (rFID 1.973 → 1.716), the largest gain. In every case GEAR improves both generation and reconstruction, confirming that the guidance mechanism is independent of the quantization scheme.

Model size. table 9 scales GEAR on top of LlamaGen-REPA at the B, L and XL sizes. As expected, generation improves monotonically with model size (gFID 21.52 → 10.63 → 7.69, IS 54.4 → 92.7 → 115.5). More surprisingly, the reconstruction quality of the *co-trained* tokenizer also improves monotonically (PSNR 20.751 → 20.808, SSIM 0.556 → 0.559 and rFID 1.658 → 1.624 from B to XL), whereas the frozen baseline tokenizer is by construction identical across sizes (rFID 1.724). Because the tokenizer is supervised only indirectly, through the soft guidance, this indicates that a stronger AR model provides a better guidance signal, yielding a tokenizer that is at once easier to sample from and higher in fidelity.

Representation encoder. table 10 replaces the alignment target. GEAR improves generation regardless of the choice of pretrained encoder, lowering gFID by 4–6 points for DINOv2 [29], DINOv3 [38], SigLIPv2 [43]

Table 10 Ablation Study on Representation Encoder. The baselines and GEAR are both trained for 100k steps.

Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
DINOv2-B [29]	22.371	6.522	52.982	0.635	**0.620**	21.061	0.565	1.724
+GEAR (Ours)	**16.837**	**5.408**	**65.098**	**0.670**	0.611	20.925	0.561	1.721
DINOv3-B [38]	23.115	6.425	51.483	0.631	**0.621**	21.061	0.565	1.724
+GEAR (Ours)	**18.967**	**5.597**	**59.838**	**0.658**	**0.621**	21.021	0.566	1.696
SigLIPv2-B [43]	23.254	6.369	50.830	0.631	**0.627**	21.061	0.565	1.724
+GEAR (Ours)	**19.226**	**5.360**	**59.047**	**0.656**	0.611	20.867	0.558	1.803
V-JEPA2.1-B [28]	23.536	6.369	50.216	0.638	0.608	21.061	0.565	1.724
+GEAR (Ours)	**19.532**	**5.876**	**56.754**	**0.653**	**0.610**	20.759	0.553	1.879

Table 11 Ablation Study on Alignment Depth. The baselines and GEAR are both trained for 100k steps.

Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
6th layer	23.102	6.428	51.912	0.636	**0.618**	21.061	0.565	1.724
+GEAR (Ours)	**16.961**	**5.331**	**64.200**	**0.673**	0.609	20.889	0.560	1.714
8th layer	22.371	6.522	52.982	0.635	**0.620**	21.061	0.565	1.724
+GEAR (Ours)	**16.837**	**5.408**	**65.098**	**0.670**	0.611	20.925	0.561	1.721
10th layer	22.455	6.417	52.299	0.633	**0.628**	21.061	0.565	1.724
+GEAR (Ours)	**17.988**	**5.380**	**61.915**	**0.666**	0.616	20.930	0.561	1.714

Table 12 Ablation Study on Tokenizer Initialization. "w/ init." starts the GEAR-L tokenizer from the warm-up checkpoint (our default), while "w/o init." trains it from scratch jointly with the AR.

Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	PSNR↑	SSIM↑	rFID↓
LlamaGen-REPA-L [23]	14.719	6.009	77.333	0.670	**0.640**	**21.061**	**0.565**	1.724
GEAR (w/o init.)	13.435	5.810	80.696	0.689	0.619	20.515	0.547	2.256
GEAR (w/ init.)	**10.630**	**5.111**	**92.700**	**0.701**	0.630	20.779	0.558	**1.640**

and V-JEPA2.1 [28]. DINOv2 gives the strongest result (16.837 gFID) and is used as our default target.

Alignment depth. table 11 varies the AR layer at which alignment is applied. GEAR improves over the corresponding baseline at every depth, and the 8-th layer attains the best gFID (16.837), which we adopt by default.

Tokenizer initialization. table 12 asks whether GEAR needs a pretrained tokenizer to start from. Initializing the end-to-end tokenizer from the warm-up checkpoint (our default) is best (gFID 10.630, rFID 1.640). Training the tokenizer *from scratch* jointly with the AR still beats the frozen pretrained baseline on generation (gFID 13.435 vs. 14.719 for LlamaGen-REPA), though its reconstruction is worse (rFID 2.256). Initialization is thus not required for the guidance to help, but it gives a sizable extra boost and keeps the tokenizer's reconstruction intact, so we initialize from the warm-up tokenizer by default.

Resolution. table 13 extends GEAR to higher generation resolutions (384 and 512). Following LlamaGen, images are sampled at the training resolution and resized to 256 before computing FID. GEAR improves generation over LlamaGen-REPA at every resolution and under both sampling settings: with CFG it lowers the gFID from 7.26 to 5.49 at 256, 7.16 to 5.95 at 384, and 10.20 to 7.60 at 512 (with consistently higher IS), confirming that the end-to-end guidance transfers to higher resolutions.

Table 13 Ablation Study on Resolution. All models are trained for 100k steps. Following LlamaGen [40], we sample at the training resolution (e.g., 384, 512) and resize to 256 before computing FID. The w/ CFG columns use a guidance scale of 1.5.

Resolution	Generation w/o CFG					Generation w/ CFG				
	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
256	22.371	6.522	52.982	0.635	**0.620**	7.255	5.587	127.827	0.788	**0.531**
+GEAR (Ours)	**16.837**	**5.408**	**65.098**	**0.670**	0.611	**5.492**	**5.167**	**157.681**	**0.822**	0.511
384	22.977	7.226	54.079	0.616	**0.631**	7.159	5.739	130.327	0.772	**0.554**
+GEAR (Ours)	**18.364**	**5.874**	**62.428**	**0.658**	0.605	**5.945**	**5.383**	**143.497**	**0.807**	0.508
512	27.742	8.720	46.125	0.587	**0.633**	10.204	6.650	107.186	0.736	**0.553**
+GEAR (Ours)	**21.826**	**6.336**	**55.689**	**0.634**	0.613	**7.595**	**5.621**	**126.386**	**0.779**	0.531

Classifier-free guidance. table 14 sweeps the CFG scale to find the best trade-off between sample quality and diversity. Generation follows the familiar pattern: increasing the scale rapidly improves gFID and IS while reducing Recall, with the best gFID of 3.388 reached at a scale of 1.5 (from 10.63 without guidance). We therefore adopt a CFG scale of 1.5 for guided sampling.

Table 14 Ablation Study on CFG Scale.

CFG Scale	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
1.00 (w/o CFG)	10.630	**5.111**	92.700	0.701	**0.630**
1.25	4.811	5.114	154.546	0.786	0.570
1.50	**3.388**	5.446	216.915	0.840	0.525
1.75	3.916	5.928	268.230	0.872	0.471
2.00	5.264	6.540	**309.665**	**0.895**	0.431

5 Discussion

Reconstruction ceiling. Under a far smaller training budget, GEAR substantially narrows the gap between autoregressive generation and strong latent-diffusion baselines, yet it still trails the best end-to-end diffusion model, REPA-E [20]. The reason is the discrete tokenizer. GEAR's reconstruction (rFID 1.64) upper-bounds its generation (gFID 2.52 with CFG), whereas REPA-E's continuous VAE reconstructs far more faithfully (rFID 0.28) and reaches gFID 1.12. Closing this reconstruction gap is the single largest lever for further improving VQ-AR generation.

Compression is coupled to compute in autoregression. The deeper cause is architectural. In current VQ-AR pipelines the tokenizer down-samples by 16×, mapping a 256×256 image to 256 tokens, and the AR model spends its compute on exactly those 256 tokens, so the compression rate and the sequence length are tied together. Latent diffusion instead decouples them. It uses a milder 8× tokenizer that keeps 1024 latent positions, and thus higher fidelity, then applies a 2×2 patch embedding so the transformer still operates on 256 tokens. Because one token is one decoding step, an AR model cannot lengthen its latent without lengthening its sequence and its compute, and is therefore pushed toward a more aggressive, lossier tokenizer. Borrowing this decoupling from the diffusion side, for instance a milder tokenizer paired with AR-side grouping such as patchified or multi-token prediction, is a promising route to raise the reconstruction ceiling without inflating the sequence.

Toward unified, long-context generation. Despite this ceiling, the next-token formulation remains attractive for several reasons. It is uniform across modalities and friendly to scaling and engineering. Its discrete tokens also bound the per-step error, whereas continuous latents accumulate it over a long context, and unified pipelines that re-encode and decode across heterogeneous understanding and generation encoders, such as BAGEL [7], compound this drift across turns. Finally, the discrete next-token form directly inherits the mature alignment stack of large language models, from reinforcement learning from human feedback [30] to preference-optimization methods such as PPO [34], DPO [32] and GRPO [35]. Such preference alignment is almost always the final and indispensable step before a model is deployed in practice. End-to-end discrete VQ-AR, as instantiated by GEAR, is therefore a promising substrate for unified, long-context understanding and generation.

6 Conclusion

We presented GEAR, a framework that trains a VQ tokenizer and an autoregressive generator jointly and end-to-end. The key idea is to let the AR model's representation-alignment objective guide the tokenizer through a differentiable soft assignment, while a hard, one-hot branch trains the generator on exactly the discrete tokens used at inference. Routing only the alignment signal, and never the prediction loss, into the tokenizer overcomes the non-differentiable index that defeats the straight-through estimator and avoids the code collapse it would otherwise induce. Across class-conditional ImageNet and text-to-image generation, GEAR substantially accelerates training relative to the strong LlamaGen-REPA baseline (up to $10\times$ faster ImageNet gFID convergence) and improves final quality, while also improving the reconstruction of the co-trained tokenizer. A representation analysis shows that these gains come from sharper patch-level, spatially-coherent structure, which is exactly what makes next-token prediction easier, and the mechanism is a drop-in across quantizers (VQVAE, LFQ, IBQ). We believe guided end-to-end training is a general principle for visual generation, and a promising next step is to scale it to larger text-to-image models and to unified understanding-and-generation systems.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[2] Keshigeyan Chandrasegaran, Kyle Sargent, Suchir Agarwal, Michael Jang, Michael Poli, Juan Carlos Niebles, Justin Johnson, Jiajun Wu, and Li Fei-Fei. Gpic: A giant permissive image corpus for visual generation. *arXiv preprint arXiv:2605.30341*, 2026.

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[4] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

[5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pages 57611–57640, 2024.

[6] Wenda Chu, Bingliang Zhang, Jiaqi Han, Yizhuo Li, Linjie Yang, Yisong Yue, and Qiushan Guo. End-to-end autoregressive image generation with 1d semantic tokenizer. *arXiv preprint arXiv:2605.00503*, 2026.

[7] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[8] Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

[9] Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, et al. Vqrae: Representation quantization autoencoders for multimodal understanding, generation and reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30322–30334, 2026.

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[11] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 23164–23173, 2023.

[12] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

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

[14] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[16] Jiaming Han, Hao Chen, Yang Zhao, Hanyu Wang, Qi Zhao, Ziyan Yang, Hao He, Xiangyu Yue, and Lu Jiang. Vision as a dialect: Unifying visual understanding and generation via text-aligned representations. *Advances in Neural Information Processing Systems*, 38:158430–158459, 2026.

[17] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[18] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[19] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[20] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18262–18272, 2025.

[21] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 36115–36125, 2026.

[22] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[23] Bin Lin, Zongjian Li, Yuwei Niu, Kaixiong Gong, Yunyang Ge, Yunlong Lin, Mingzhe Zheng, JianWei Zhang, Miles Yang, Zhao Zhong, et al. ifsq: Improving fsq for image generation with 1 line of code. *arXiv preprint arXiv:2601.17124*, 2026.

[24] Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, et al. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

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

[26] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[27] Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

[28] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[30] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

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

[35] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[36] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16037–16046, 2025.

[37] Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. *arXiv preprint arXiv:2603.14366*, 2026.

[38] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[39] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *arXiv preprint arXiv:2512.10794*, 2025.

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

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

[42] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[43] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[44] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[45] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[46] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[47] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[48] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[49] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14273–14282, 2026.

[50] Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, et al. What matters for diffusion-friendly latent manifold? prior-aligned autoencoders for latent diffusion. *arXiv preprint arXiv:2605.07915*, 2026.

[51] 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 conference on computer vision and pattern recognition*, pages 586–595, 2018.

[52] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

A Ablation Training Configurations

All ablation studies fine-tune the GEAR-L model end-to-end on top of the warm-up tokenizer (except the initialization study, which also trains the tokenizer from scratch), and each study varies a single axis while holding everything else fixed. table 15 gives one column per study, with the axis each study sweeps shown in **bold** (default underlined). The optimization, loss-weight and sampling settings shared by all studies are listed separately in table 16.

A few conventions deserve a note. The alignment target is spatially normalized following iREPA [39] (per-channel z-score with strength α=0.6) only for the SigLIPv2 and V-JEPA2.1 targets, which need it to serve as good REPA targets. DINOv2 and DINOv3 use no spatial normalization, as in the original REPA [48]. Across model sizes the alignment is always applied at one-third of the depth: the B/L/XL backbones have 12/24/36 layers, so the alignment layer is 4/8/12 respectively, which is the optimal alignment depth identified by LlamaGen-REPA [23]. The reconstruction loss weights are read off the tokenizer objective of equation (2).

Table 15 Per-study training configurations for the ablation studies (Tables 5–13). Each column is one study and lists its configuration, with the axis it sweeps in **bold** (default underlined). Settings shared by all studies are listed in table 16.

Setting	Temp. Tab. 5	Coeff. Tab. 6	Comp. Tab. 7	Tok. Tab. 8	Size Tab. 9	Enc. Tab. 10	Depth Tab. 11	Init Tab. 12	Res. Tab. 13
AR model	L	L	L	L	**B** **L** **XL**	L	L	L	L
Alignment layer ℓ	8	8	8	8	4 8 12	8	**6** **8** **10**	8	8
REPA target	DINOv2	DINOv2	DINOv2	DINOv2	DINOv2	**DINOv2** **DINOv3** **SigLIPv2** **V-JEPA2.1**	DINOv2	DINOv2	DINOv2
Tokenizer	VQ-16	VQ-16	VQ-16	**VQ-16** **LFQ-16** **IBQ-16**	VQ-16	VQ-16	VQ-16	VQ-16	VQ-16
Gradient	soft	soft	**soft** **STE** **w/o $\mathcal{L}_{\text{GAN}}$**	soft	soft	soft	soft	soft	soft
Guidance temp. τ	**0.5** **0.1** **0.05** **0.01** **0.005**	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Align. coeff. λ	0.5	**0.25** **0.5** **0.75** **1.0**	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Resolution	256	256	256	256	256	256	256	256	**256** **384** **512**
Training steps	400k	400k	400k	400k	400k	100k	100k	400k	100k
Tokenizer init.	✓	✓	✓	✓	✓	✓	✓	✗	✓

Table 16 Shared training configuration (Tables 5--13 and Tables 1--2). Optimization, loss-weight and sampling settings used by every ablation study.

Batch size	256	Precision	bf16
Learning rate	1×10^{-4}	Codebook size	16384
Optimizer	AdamW	CLS dropout	0.1
Adam β_1	0.9	Projector dim	2048
Adam β_2	0.999	EMA decay	0.9999
Weight decay	0	torch.compile	✓
Gradient clip	1.0	Entropy	0.05
NTP loss	1.0	Commitment	0.25
Reconstruction (L1)	1.0	Sampling temperature	1.0
Perceptual (LPIPS)	0.1	Top-k	0
Adversarial (GAN)	0.1	Top-p	1.0

Table 17 Text-to-image training configuration on GPIC. GEAR and LlamaGen-REPA share this recipe and differ only in whether the frozen tokenizer has been end-to-end fine-tuned. The model is trained from scratch, and the LlamaGen-1B backbone matches the parameter count of the GPIC JiT-1B baseline.

Frozen VQ tokenizer. **GEAR**: warm-up, then end-to-end fine-tuned. **LlamaGen-REPA**: warm-up only.			
AR model	LlamaGen-1B	AR initialization	**from scratch**
Align layer	12	Resolution	256
Training data	GPIC (∼100M)	Schedule	1 epoch (∼390k)
LR schedule	constant	AR learning rate	1×10^{-4}
Batch size	256	Text encoder	Qwen3-1.7B
Text length	300	Caption dropout	0.1
Attention	causal	REPA target	DINOv2
Align. coeff. λ	0.5	Optimizer	AdamW (0.9, 0.999)
Weight decay	0	Gradient clip	1.0
Precision	bf16	Codebook size	16384
EMA decay	0.9999	torch.compile	✓
Sampling temperature	1.0	Top-k	0
Top-p	1.0		

B Text-to-Image Training Configurations

Our text-to-image model is a strict autoregressor over the concatenation of the text and image tokens. Following an MMDiT-style hybrid design, the first third of the transformer blocks are *dual-stream* (separate query/key/value/output projections for the text and image tokens) and the remaining two thirds are *single-stream* with shared projections, while the attention stays purely causal throughout. The text condition is a 300-token Qwen3-1.7B embedding with a 0.1 caption-dropping probability for classifier-free guidance, and a DINOv2 REPA loss is applied as in the class-conditional setting.

The model is trained **from scratch** on a LlamaGen-1B backbone, which matches the parameter count of the GPIC JiT-1B baseline we compare against, for a single epoch over the ∼100M-image GPIC corpus at 256 resolution (about 390k steps at batch size 256) with a constant learning rate. Since there is no class-conditional checkpoint to inherit, the text stream is learned from scratch as well. GEAR and LlamaGen-REPA follow this identical recipe and differ only in whether the frozen tokenizer has been end-to-end fine-tuned. table 17 lists the configuration.

Table 18 Tokenizer reconstruction on the ImageNet validation set. *Setting* is the source of each row: *Reported* from the official paper, *Reproduced* our re-evaluation of the released weights, *Warm-up* after recovering the GAN discriminator, and *GEAR* after end-to-end training (the last two repeat table 8). We evaluate with bicubic, whereas the official LFQ/IBQ numbers use bilinear. *LlamaGen's official SSIM is computed treating $[0, 1]$ images as $[-1, 1]$ and is therefore inflated.

Tokenizer	Setting	Interp.	Params	Codebook	Dim	rFID↓	PSNR↑	SSIM↑	L1↓
VQ-16 [40]	Reported	–	71.9M	16384	8	2.19	20.79	0.67*	–
	Reproduced	bilinear				2.10	21.51	0.59	0.0585
	Reproduced	bicubic				2.19	20.79	0.55	0.0633
	Warm-up	bicubic				1.72	21.06	0.57	–
	GEAR	bicubic				1.64	20.78	0.56	–
LFQ-16 [25]	Reported	bilinear	115.1M	16384	14	2.55	22.21	0.62	–
	Reproduced	bilinear				2.56	22.25	0.61	0.0524
	Reproduced	bicubic				2.82	21.47	0.58	0.0571
	Warm-up	bicubic				2.42	20.97	0.56	–
	GEAR	bicubic				2.13	20.48	0.55	–
IBQ-16 [36]	Reported	bilinear	111.0M	16384	256	2.06	22.01	0.61	–
	Reproduced	bilinear				2.05	22.04	0.61	0.0542
	Reproduced	bicubic				2.23	21.23	0.58	0.0593
	Warm-up	bicubic				1.97	21.18	0.58	–
	GEAR	bicubic				1.72	20.92	0.57	–

C Tokenizer Reconstruction and Evaluation Interpolation

We instantiate GEAR on three publicly released tokenizers, all with a 16384-entry codebook: VQ-16 from LlamaGen [40], LFQ-16 from Open-MAGVIT2 [25], and IBQ-16 [36]. table 18 reports their sizes and reconstruction quality on the ImageNet validation set.

One pipeline detail materially affects these numbers: the interpolation used when a validation image is short-side-resized and center-cropped to the evaluation resolution before being fed to the tokenizer. Since PSNR, SSIM and rFID are all computed against this resized image, the interpolation changes the reference itself. Bicubic keeps more high-frequency content (sharper) while bilinear is smoother, so the two give different scores. We evaluate with **bicubic**, whereas the official LFQ and IBQ numbers are reported with bilinear. Re-evaluating the released weights with bilinear reproduces the official numbers closely (for example, LFQ rFID 2.55 versus our 2.56), confirming that the gap reflects the interpolation rather than a different checkpoint. The *warm-up* and *GEAR* rows repeat the reconstruction metrics from table 8 (also measured with bicubic), showing that recovering the GAN discriminator and then training end-to-end preserves reconstruction. The official LlamaGen SSIM (marked *) is inflated because its implementation computes SSIM treating $[0, 1]$-valued images as $[-1, 1]$.

D Classifier-Free Guidance Sweep for Text-to-Image

Under the controlled GPIC setting (tables 2 and 3) we sweep the classifier-free guidance (CFG) scale from 1 to 20 on DPG-Bench and on GenEval (short and long prompts), for both the non-end-to-end tokenizer (LlamaGen-REPA) and the end-to-end one (GEAR). table 19 reports the full sweep.

Observations. The end-to-end tokenizer (GEAR) is the stronger of the two across essentially the entire sweep. On DPG-Bench the two are within noise at very low guidance (CFG ≤ 2), but from CFG ≥ 4 GEAR pulls ahead by a stable $+1.3$ to $+2.5$ points. Its best score is 72.881 at CFG=16, against 71.229 at CFG=18 for the non-e2e tokenizer ($+1.65$ best-vs-best), reaching the higher peak at a *smaller* guidance scale. On GenEval the advantage holds at every scale and for both prompt sets, with a roughly constant gap of $\approx +0.05$ (short) and $\approx +0.06$ (long). The long-prompt scores peak near CFG=16, whereas the short-prompt scores are still climbing at CFG=20.

Discussion. A salient feature of these curves is that quality keeps rising up to very large guidance scales. A well-fit text-to-image model rarely needs a CFG of 16–20, so the fact that performance is still improving

Table 19 Classifier-free guidance sweep on the GPIC-trained text-to-image models. Each block reports one benchmark for the *non-e2e* tokenizer (LlamaGen-REPA) and the *e2e* tokenizer (GEAR) across CFG scales 1–20. GenEval is shown for its original (short) and LLM-refined (long) prompts. The best of each row is in **bold** and the runner-up is underlined.

Benchmark	Tokenizer	Classifier-free guidance scale											
		1.0	1.75	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0	18.0	20.0
DPG-Bench	non-e2e	55.386	64.524	64.928	68.357	69.927	70.118	70.634	70.336	71.017	70.363	**71.229**	71.062
	e2e	55.369	63.782	64.815	69.199	71.281	71.511	72.134	72.383	72.857	**72.881**	72.753	72.806
GenEval (short)	non-e2e	0.0743	0.1400	0.1661	0.2481	0.2443	0.2572	0.2617	0.2739	0.2708	0.2717	0.2814	**0.2832**
	e2e	0.0861	0.1627	0.1862	0.2722	0.2964	0.3184	0.3149	0.3269	0.3245	0.3339	0.3295	**0.3413**
GenEval (long)	non-e2e	0.2181	0.3117	0.3284	0.3817	0.3854	0.4028	0.4164	0.4096	0.4163	**0.4187**	0.4101	0.4182
	e2e	0.2269	0.3401	0.3605	0.4228	0.4431	0.4595	0.4579	0.4767	0.4601	**0.4779**	0.4701	0.4705

there indicates that the single-epoch GPIC model is *underfitting* and does not yet exploit the text condition strongly. This is expected given the deliberately small training budget. We stress that the goal of this study is a strictly controlled comparison that isolates the effect of the (end-to-end) tokenizer, not to chase state of the art: under this matched recipe GEAR's end-to-end tokenizer is consistently better across the whole guidance range and reaches higher peaks at smaller CFG, mirroring the class-conditional results. We expect this advantage to carry over once the recipe is scaled with longer training and stronger text conditioning, which we leave to future work.

E Per-Layer Representation Analysis

In figure 5, the image-level similarities (a,b) and Moran's I (c) peak near the alignment depth (ℓ=8), whereas the patch-level curves (d,e) peak only in deeper layers. This is expected rather than anomalous. Image-level similarity reflects the global semantics that REPA injects directly at layer 8. Patch-level similarity instead reflects intra-image spatial structure, which only takes shape once deeper, causal layers have accumulated enough spatial context. We also probe the raw, pre-projection features, so the alignment layer need not be the most DINO-like in raw space. The bump in the FFT high-frequency ratio (f) at ℓ=8 has the same origin: REPA constrains only the *projected* feature $g_\phi(\mathbf{h}^{(8)})$ by cosine direction, so the raw alignment-layer feature is free to carry the injected per-token semantics in a high-frequency form, which the deeper causal layers then smooth into the spatially coherent, DINOv2-like structure that actually drives generation. Importantly, at *every* depth GEAR dominates LlamaGen-REPA at the patch level and in both locality statistics (including this bump), which is what matters for generation.