ADVERSARIAL ERROR CORRECTION FOR VISUAL AUTOREGRESSIVE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual Autoregressive (VAR) models have emerged as a powerful paradigm for image synthesis by performing hierarchical next-scale prediction. However, VAR models are inherently prone to cascading error propagation, where subtle coarse-scale mispredictions are amplified across the hierarchy, ultimately distorting the final synthesis. To mitigate this, we propose AID-VAR, a plug-and-play framework that enhances pre-trained VARs through Adversarially Injected Diagnosis. Instead of a standard passive generation, AID-VAR introduces a proactive error-correction mechanism inspired by the adversarial feedback in GANs. We deploy a discriminator to diagnose fidelity gaps at each scale transition, coupled with a lightweight guidance injector. This module operates as a non-invasive adapter that refines the feature manifold of a frozen VAR backbone, effectively steering the generation toward the distribution of real images without destabilizing the pre-trained latent space. Furthermore, to rigorously evaluate this cross-scale progression, we introduce the Inter-Scale Consistency Score (ISCS), a novel metric that quantifies the fidelity and structural alignment between consecutive resolution scales. Experimental results across various backbones demonstrate that AID-VAR delivers sharper textural details and fewer structural distortions with negligible overhead. For instance, AID-VAR-d20 achieves a 16% improvement in FID with only a 3% increase in parameters. These results establish AID-VAR as a highly efficient and scalable pathway for upgrading large-scale VAR generators—enhancing global coherence and local detail without altering training data, base architectures, or sampling schedules.

1 INTRODUCTION

Autoregressive modeling has recently emerged as a competitive paradigm for high-fidelity image synthesis, offering stable training, exact likelihood, and flexible conditioning Van Den Oord et al. (2016; 2017); Esser et al. (2021); Chang et al. (2022); Tian et al. (2024). In the visual autoregressive (VAR) Tian et al. (2024) setting, an image is synthesized across a hierarchy of spatial scales; each stage conditions on previously generated content and predicts the next chunk of visual information. Despite their strong generative capacity, VARs suffer from a characteristic failure mode: local prediction errors accumulate across scales, compounding into detail loss, texture drift, and small-region distortions that undermine global coherence, as illustrated in Figure 1 and Figure 2. These issues stem from exposure bias Bengio et al. (2015); Ranzato et al. (2015); Schmidt (2019); He et al. (2019) and imperfect cross-scale transitions: once a local mistake enters the chain, subsequent decoders amplify rather than correct it Parthipan et al. (2024); Pasini et al. (2024); Lee et al. (2022); Tian et al. (2024); Cheng et al. (2025); Lin et al. (2024); Yue et al. (2025).

We argue that mitigating these cross-scale errors requires diagnostic feedback at each scale transition, capable of identifying "off-manifold" artifacts and nudging the generation back toward the natural image distribution. Adversarial learning Goodfellow et al. (2014); Brock et al. (2018); Karras et al. (2019; 2021) provides precisely such a signal: a discriminator trained on real *v.s.* generated images excels at detecting fine-grained visual pathologies (*e.g.*, broken micro-geometry, aliasing, texture misalignment). However, naïvely adversarially fine-tuning the base VAR is often impractical: it introduces training instability, alters a large number of parameters, and demands substantial compute to retain the original model's strengths.

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.

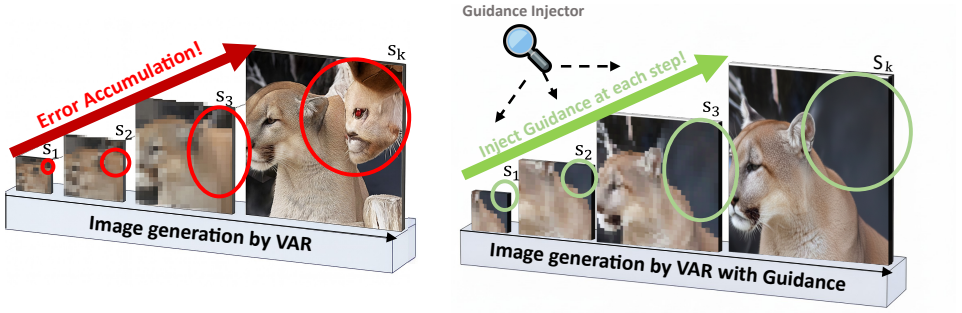


Figure 1: Conceptual illustration of our motivation. **Left:** The standard VAR model's coarse-to-fine generation is prone to error accumulation, where initial inaccuracies (circled in red) are magnified across scales, leading to severe structural degradation. **Right:** Our AID-VAR framework introduces a guidance injector at each step to anticipate and correct these errors, ensuring a globally coherent and plausible final image.

This paper introduces AID-VAR (Adversarially Injected Diagnosis for VAR), a lightweight, plug-and-play guidance module that upgrades a pre-trained, frozen VAR with adversarial feedback, without modifying training data, the base architecture, or the sampling schedule. AID-VAR first trains a discriminator to detect visual errors and leverages it to supervise a small guidance injector attached to the VAR. During generation, the injector conditions on previously generated scales and injects a spatial guidance signal that steers the next scale's prediction toward the natural-image manifold identified by the discriminator. Crucially, the base VAR remains frozen; only the tiny injector and the discriminator heads are trained, yielding stability, efficiency, and easy implementation across backbones.

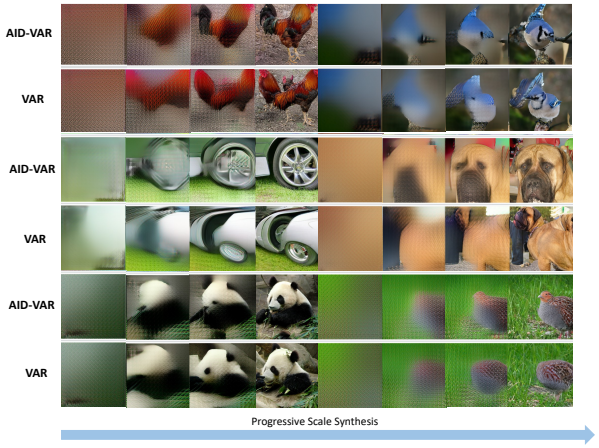


Figure 2: Qualitative comparison between our AID-VAR and the standard VAR across generation scales. The comparison demonstrates that AID-VAR better preserves visual consistency and reduces error accumulation during progressive generation.

To implement the above error-fixing mechanism via adversarial guidance, a key question is where the discriminator should operate. While the VAR's native latent space, where multi-scale tokens reside, seems natural, our experiments show that discriminators trained directly in this compressed space suffer from rapid collapse and weak gradients. Latent representations, though semantically rich, lack the explicit spatial and textural details needed to discriminate subtle visual artifacts. We therefore propose a differentiable pipeline that operates in RGB pixel space rather than latent space. Specifically, we employ a projected discriminator Sauer et al. (2021) that leverages a fixed, pretrained image encoder to extract robust perceptual features and learns only shallow classification heads on top. To bridge the gap between RGB-space supervision and latent-space optimization, we introduce soft-label based VQ-VAE decoding. Unlike standard hard quantization that blocks gradients, our soft decoding computes continuous weighted combinations of codebook vectors, providing stable gradients from pixel space back through the decoder to the injector. This maintains end-to-end differentiability while preserving the discrete structure of the VAR. Finally, to prevent the discriminator from overfitting to stale artifacts, we adopt a dynamic training refresh strategy. At each training iteration: (1) the frozen VAR paired with the current injector generates refined images via the soft-decoding path; (2) the discriminator is updated on real images versus these current refined samples; (3) the injector is updated using the discriminator's adversarial signal. This on-the-fly refresh stabilizes the min-max game, reduces overfitting to outdated generation patterns, and improves sample efficiency.

We further observe that hierarchical autoregressive generation calls for evaluation beyond final-image realism. Standard image-level metrics, including FID Heusel et al. (2017), IS Salimans et al. (2016), and precision/recall Kynkäänniemi et al. (2019), mainly measure the quality of completed images,

but do not reveal whether each refinement stage remains faithful to its coarse-scale context. To fill this gap, we introduce the Inter-Scale Consistency Score (ISCS), which measures the fidelity of transitions between consecutive scales. ISCS serves as a targeted probe for exposure bias across the hierarchy: higher ISCS indicates that fine-scale refinements remain consistent with coarse-scale structure, aligning with perceptual coherence. Evaluated under ISCS, baseline VARs score low due to sensitivity to early errors and semantic drift; in contrast, our AID-VAR substantially boosts ISCS, indicating stabilized cross-scale refinement and reduced error propagation. Our main contributions are summarized as follows:

- We propose a novel adversarial framework to address the cumulated error issue in VAR, by diagnosing and correcting cross-scale errors via a lightweight injector.
- We make adversarial supervision effective and stable for VARs through an RGB-space discriminator and a differentiable soft-decode route that jointly deliver informative gradients to the injector, together with a dynamically refreshed training loop that maintains up-to-date fake samples.
- We introduce ISCS to quantify cross-scale coherence, complementing FID/IS by directly targeting exposure-bias effects between adjacent scales.
- Extensive experiments on ImageNet-1K Deng et al. (2009) demonstrate that AID-VAR consistently improves multiple VAR backbones while introducing only modest additional overhead.

2 Related Work

Autoregressive models. Inspired by their success in NLP (Brown et al., 2020; Radford et al., 2018; Devlin et al., 2019), autoregressive (AR) models have been widely explored for image generation. Early methods relied on next-pixel prediction over raster-scanned sequences (Van Den Oord et al., 2016; Van den Oord et al., 2016). The introduction of visual tokenizers such as VQ-VAE (Van Den Oord et al., 2017) and VQGAN (Esser et al., 2021) shifted generation toward next-token prediction (Ramesh et al., 2021; Yu et al., 2022; 2021), substantially improving image quality but requiring many decoding steps. Visual Autoregressive Modeling (VAR) (Tian et al., 2024) alleviated this bottleneck through a "next-scale prediction" paradigm, while other works explored non-sequential or parallel decoding (Chang et al., 2022; Li et al., 2023; Tschannen et al., 2024). Nevertheless, AR generation remains vulnerable to error accumulation due to its sequential dependency structure Bengio et al. (2015); Ranzato et al. (2015); Schmidt (2019); Parthipan et al. (2024); He et al. (2019); Pasini et al. (2024); Tian et al. (2024); Cheng et al. (2025); Lin et al. (2024). AID-VAR addresses this issue by introducing external guidance to correct the generation process without modifying the base model.

Adversarial training for generative models. Since the introduction of GANs (Goodfellow et al., 2014), adversarial training has been widely used both for standalone generators (Radford et al., 2015; Karras et al., 2019; Brock et al., 2018; Zhang et al., 2019) and for enhancing other generative models, including diffusion models (Dhariwal & Nichol, 2021; Ho et al., 2020; Song et al., 2020; Rombach et al., 2022). Adversarial objectives have been applied to improve perceptual quality, accelerate sampling, and distill pre-trained models (Rombach et al., 2022; Lu et al., 2022; Salimans & Ho, 2022; Sauer et al., 2024). Our training strategy is particularly inspired by Adversarial Diffusion Distillation (ADD) (Sauer et al., 2024) and Projected GANs (Sauer et al., 2021), which use frozen pre-trained feature extractors with lightweight trainable discriminator heads. Different from prior work that applies adversarial objectives to the main generator, AID-VAR uses this framework solely to train an independent lightweight guidance injector, enabling correction of a frozen pre-trained model without costly end-to-end adversarial training.

3 AID-VAR for Error Diagnosis

Given a frozen VAR that predicts multi-scale tokens, we aim to reduce cross-scale error accumulation without modifying its weights, data, or sampling schedule. AID-VAR introduces a lightweight *guidance injector*, trained adversarially with a projected RGB-space *discriminator*, to produce spatial guidance from previous scales and inject it into the next-scale prediction. The overall framework is shown in Figure 3.

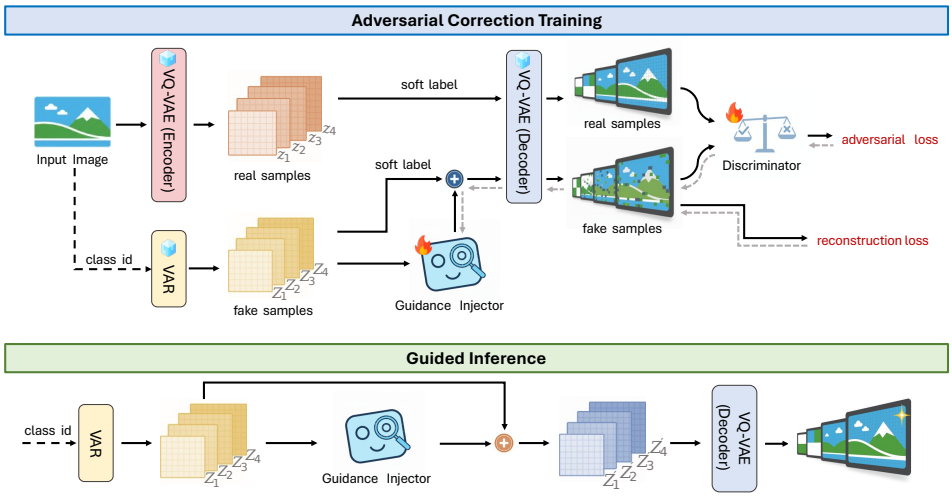


Figure 3: **AID-VAR framework for guided autoregressive generation.** (1) **Training:** We freeze the pretrained VAR and learn a discriminator to adversarially train a lightweight guidance injector, which produces spatial guidance maps that are injected into the VAR's next-scale predictions via a differentiable soft-decoding path. (2) **Inference:** Only the guidance injector is attached to the frozen VAR to provide supplemental guidance signals, yielding a simple, efficient drop-in upgrade of VAR.

3.1 Learning to discriminate scales

Preliminary: next-scale prediction in VARs. Visual autoregressive models synthesize an image over a coarse-to-fine hierarchy of scales. At scale k, the model predicts the next block of visual tokens conditioned on all previously generated tokens (and any optional external condition c, *e.g.*, class/text). Let S_k denote the predicted tokens at scale k, x_k the corresponding hidden state, and z_k the pre-softmax logits. A generic VAR defines

$$z_k = f_{\text{VAR}}(x_k | \{S_{<k}\}), \quad S_k = \arg\max \text{softmax}(z_k), \tag{1}$$

where each x_k is conditioned on $\{S_{<k}\}$ and c. The process proceeds from coarse to fine and generates K intermediate scales; after the finest scale, tokens are decoded to an RGB image by a VQ-VAE decoder (Van Den Oord et al., 2017). However, this sequential generation process over discrete tokens inevitably leads to severe error accumulation, a crucial limitation we formally prove in Appendix A.1.

Learning to discriminate intermediate scale errors. We seek a discriminator that is maximally sensitive to fine-grained visual defects that emerge in the generated intermediate scales, yet stable enough to supervise a lightweight guidance injector without updating the frozen VAR (as theoretically analyzed in Appendix A.3).

Therefore, following ADD (Sauer et al., 2024), a more promising way to construct the discriminator is leveraging a pre-trained visual foundation model (*e.g.*, DINO (Caron et al., 2021), CLIP (Radford et al., 2021)) that generalize well to the image samples. Formally, as the architecture shown in Figure D.1(b), given an input image I, a frozen, pretrained vision encoder E extracts features, which feed into shallow, learnable heads h:

$$D(I; \theta) := h(E(I); \theta), \tag{2}$$

where only θ is trained, and E is frozen. We empirically find this DINO-based discriminator enjoys multiple benefits compared to randomly-initialized discriminator:

- Perceptual alignment: pretrained encoders in RGB capture natural-image statistics that are directly predictive of aliasing, texture drift, and micro-geometry breaks.
- Stability: discriminating in discrete/token spaces introduces distributional mismatch and early collapse, whereas RGB features are smoother and more linearly separable for real/fake cues.
- plug-and-play: RGB-space supervision is agnostic to tokenizer/codebook design and generalizes across VAR backbones.

Training data. The discriminator itself can be regarded as a binary classifier to identify real and fake samples. So the training data of discriminator is composed with real samples and fake (generated)

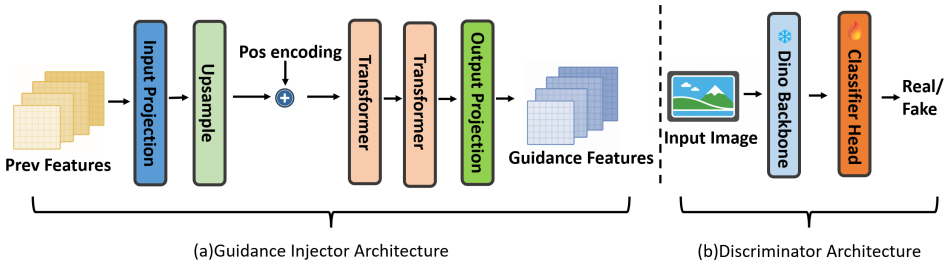


Figure 4: Architectures for the (a) guidance injector and (b) discriminator. The injector is a lightweight Transformer encoder that processes previous scale features. The discriminator uses a frozen DINO backbone with a trainable classification head.

samples: (1) Real set: We sample the image I uniformly from the training dataset of VAR, then feeds it into the VAR encoder to obtain tokens at each scale k, and use the decoder to obtain the corresponding images I_k at each scale. (2) Fake set: we run the inference of the frozen VAR to generate next-scale tokens S_k at each scale k. We then obtain the RGB images $\hat{I}_k$ via the decoder.

Objective. We use hinge loss (Lim & Ye, 2017) to train the discriminator on real and fake samples:

$$\mathcal{L}_D = \frac{1}{K}\sum_{k=1}^{K}\Big\{\mathbb{E}_{I_k}\big[\max(0, 1 - D(I_k;\theta))\big] + \mathbb{E}_{\hat{I}_k}\big[\max(0, 1 + D(\hat{I}_k;\theta))\big]\Big\} \tag{3}$$

As a result, we now have a discriminator that can identify real images and fake generated images at each scale. Next, we will discuss how we use the discriminator to adversarially force the VAR to precisely generate images that lie in the manifold of real images.

3.2 Adversarial Next-Scale Prediction

With a discriminator, the traditional adversarial approach updates the generator to fool a discriminator. Applied to VARs, this entails backpropagating through a large backbone and modifying millions (or billions) of weights, which (1) incurs substantial computational overhead; (2) destabilizes training under a min–max objective; and (3) risks irrecoverable drift in pretrained knowledge (catastrophic forgetting, degraded generalization). To avoid these pitfalls, AID-VAR is implemented as a lightweight extension to a frozen VAR: we keep **all** VAR parameters unchanged and learn only a small guidance module that nudges the next-scale prediction.

Architecture of the guidance injector. The guidance injector, P_ϕ, is designed as a lightweight, spatially-aware module, as illustrated in Figure D.1(a). It takes the feature map from the previously generated scale as input. The architecture consists of an input projection layer, an upsampling block to match the spatial dimensions of the next scale, followed by the addition of positional encodings. The core of the module is a series of two standard Transformer blocks (Vaswani et al., 2017), which process the spatially-rich features. Finally, an output projection layer produces the guidance feature map G_k, which has the same dimensions as the VAR's internal hidden state for the current scale. The predicted logits z_k of each scale thus becomes:

$$z'_k = f_{\text{VAR}}(x_k + w \cdot G_k | \{S_{<k}\}), \tag{4}$$

where w is the guidance weight that controls the strength of the injected guidance signal G_k.

Soft-label decoding for differentiability. The VQ-VAE path in typical VARs is discrete, in which the direct token selection $S_k = \arg\max \text{softmax}(z_k)$ is non-differentiable. To optimize P_ϕ with adversarial signals in RGB space, we introduce a soft-label decoding route that preserves gradients from image space back to z'_k. The new decoding replaces the original discrete codebook lookup operation $H = W_{[S_k]}$ with

$$H' = W \cdot \text{softmax}(z'_k), \tag{5}$$

where W is the codebook embedding matrix. This soft reconstruction enables a discriminator to act in RGB while providing stable gradients to the injector.

Note that to prevent the discriminator from exploiting reconstruction artifacts specific to the decoding route, we match its input pipeline for both real and generated samples in training data. This means both real and generated samples use soft-label decoding to obtain the images fed into the discriminator.

Adversarial loss. Similar to the adversarial loss in Equation (3), the injector at each scale k is guided with a hinge-style adversarial loss:

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\hat{I}'_k}[D(\hat{I}'; \theta)], \tag{6}$$

where $\hat{I}'_k$ is the predicted image with guidance injector. To prevent over-aggressive deviation from the VAR prior and stabilize training, we add the token-level reconstruction loss $\mathcal{L}_{\text{rec}}$ in VAR as an auxiliary objective. The overall training objective for the guidance injector P_ϕ thus becomes:

$$\mathcal{L}_P = \mathcal{L}_{\text{adv}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}, \quad \text{where} \quad \mathcal{L}_{\text{rec}} = \mathbb{E}_k[\text{CrossEntropy}(z'_k, s_k)]. \tag{7}$$

$\mathcal{L}_{\text{rec}}$ measures the cross-entropy between the guided logits z'_k and ground-truth tokens s_k, and λ_{rec} is a balancing hyperparameter.

3.3 Guided inference

AID-VAR's inference follows a guided autoregressive procedure. The initial scale ($k = 1$) is generated using standard VAR sampling with Classifier-Free Guidance (CFG) (Ho & Salimans, 2022). For all subsequent scales ($k > 1$), the guidance injector dynamically generates a guidance map G_k from the previously sampled tokens S_{k-1}. This map is then injected into the VAR's internal state with a small, fixed weight w before the current scale's tokens are sampled. This iterative, guided process continues until the final scale, after which the VQVAE decodes the complete token sequence into the final image.

4 Inter-Scale Consistency Score (ISCS)

The phenomenon of error accumulation in VAR models can be characterized as a scale-by-scale exposure bias (Ranzato et al., 2015; Bengio et al., 2015; Schmidt, 2019; He et al., 2019; Parthipan et al., 2024; Pasini et al., 2024), where the model is conditioned on its own, potentially flawed, outputs during inference. To specifically quantify this, we introduce the Inter-Scale Consistency Score (ISCS), a novel metric designed to measure the fidelity of the generative transitions between consecutive scales. Our core hypothesis is that a model's ability to progressively refine details and correct artifacts at finer scales is a critical indicator of its overall quality. Existing metrics like FID (Heusel et al., 2017) do not isolate this crucial aspect of refinement capability across the generative hierarchy.

To address this, we propose ISCS, which measures the Fréchet distance between the joint distributions of adjacent scales. For a large set of real and generated images, we construct two corresponding sets of joint feature vectors for each scale transition $k-1 \to k$. Using a pre-trained DINO model (Caron et al., 2021) as a feature extractor, $\text{feat}(\cdot)$, we define the real joint distribution features J^k_{real} and the generated joint distribution features J^k_{gen} as:

$$J^k_{\text{real}} = \{[\text{feat}(r_{k-1}),\ \text{feat}(r_k)]\}, \quad J^k_{\text{gen}} = \{[\text{feat}(r'_{k-1}),\ \text{feat}(r'_k)]\}. \tag{8}$$

The per-scale score is the inverse of the Fréchet distance (FD) between these two sets of features:

$$\text{ISCS}_k = \frac{1}{\text{FD}(J^k_{\text{real}}, J^k_{\text{gen}})} \tag{9}$$

A higher score indicates a smaller distance and thus better consistency, which represents a smoother and more flawless transition from a coarser scale to a finer one. The final score is a weighted sum of the per-scale scores:

$$\text{ISCS} = \sum_{k=1}^{K} \alpha_k \cdot \text{ISCS}_k \tag{10}$$

To emphasize late-stage generation, we use exponential weights $\alpha_k \propto 2^k$, assigning larger weights to finer scales. This biases the metric toward high-fidelity detail synthesis and artifact removal in the final refinement steps, as shown in Figure 7b, later stages are more informative and critical.

Table 1: Quantitative results on the ImageNet 256×256 validation set, comparing our proposed AID-VAR with its unguided counterparts and representative state-of-the-art generative models. Metrics include Fréchet inception distance (FID), inception score (IS), precision (Pre) and recall (Rec). '↓' indicates lower is better, while '↑' indicates higher is better.

Type	Model	Params	Steps↓	FID↓	IS↑	Pre↑	Rec↑
GAN	BigGAN (Brock et al., 2018)	112M	1	6.95	224.5	0.89	0.38
	GigaGAN (Kang et al., 2023)	569M	1	3.45	225.5	0.84	0.61
	StyleGAN-XL (Sauer et al., 2022)	166M	1	2.30	265.1	0.78	0.53
Diffusion	ADM (Dhariwal & Nichol, 2021)	554M	250+	10.94	101.0	0.69	0.63
	LDM-4 (Rombach et al., 2022)	400M	250+	3.60	247.7	-	-
	DiT-XL/2 (Peebles & Xie, 2023)	675M	250	2.27	278.2	0.83	0.57
	L-DiT-7B (Alpha-VLLM, 2024)	7B	250	2.28	316.2	0.83	0.58
Autoregressive	VQGAN (Esser et al., 2021)	1.4B	256	15.78	74.3	-	-
	RQ-Transformer (Lee et al., 2022)	3.8B	68	7.55	134.0	-	-
	MaskGIT (Chang et al., 2022)	227M	8	6.18	182.1	0.80	0.51
	MAGVIT-v2 (Yu et al., 2023)	307M	-	3.65	200.5	-	-
	MAR (Li et al., 2024)	943M	-	2.35	227.8	0.79	0.62
	VAR-d16 (Tian et al., 2024)	310M	10	3.55	274.4	0.84	0.51
	AID-VAR-d16 (Ours)	321M	10	**3.24**	**280.0**	**0.85**	**0.51**
	VAR-d20 (Tian et al., 2024)	600M	10	2.95	302.6	0.83	0.56
	AID-VAR-d20 (Ours)	619M	10	**2.54**	**309.4**	**0.83**	**0.56**
	VAR-d24 (Tian et al., 2024)	1.0B	10	2.33	312.9	0.82	0.59
	AID-VAR-d24 (Ours)	1.02B	10	**2.08**	**316.3**	**0.82**	**0.60**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

We conduct our training and evaluation on the ImageNet-1K dataset (Deng et al., 2009). All images are processed at a resolution of 256x256. The training is performed on the official training split, and all results are reported on the validation split. To ensure a fair and direct comparison, all experimental results presented are based on the same set of generated images using a fixed random seed and the other hyper parameters. This allows for a causal and related analysis between the outcomes of each experiment.

Evaluation metrics. We evaluate the performance of our method using standard image generation metrics: Fréchet Inception Distance (FID) (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), Precision, Recall (Kynkäänniemi et al., 2019; Sajjadi et al., 2018), and the ISCS Score that we presented.

Implementation details. Our experiments are built upon three pre-trained and frozen VAR models of varying sizes: VAR-d16 (310M parameters), VAR-d20 (600M), and VAR-d24 (1.0B) (Tian et al., 2024). The trainable Guidance Injector is a lightweight Transformer with 2 layers and 8 attention heads. The discriminator is based on the StyleGAN-T architecture (Sauer et al., 2023), using a frozen DINO ViT (Caron et al., 2021) as its feature extraction backbone. The Guidance Injector and the discriminator are both trained with a learning rate of 1e-6. For the composite loss of the Guidance Injector, the reconstruction loss weight is set to $\lambda_{\text{rec}} = 0.01$. The guidance weight during the training phase is fixed at $w = 0.001$; further implementation details are provided in Appendix D.

5.2 PERFORMANCE ANALYSIS

We first present a comprehensive performance analysis of our proposed AID-VAR framework. We evaluate our method on the ImageNet 256×256 validation set and compare it against two primary sets of baselines: (1) the original, unguided VAR models of corresponding sizes, to directly measure the improvements conferred by our plug-and-play module, and (2) a wide array of state-of-the-art generative models, including leading GANs, diffusion models, and other autoregressive architectures.



Figure 5: Qualitative comparison of AID-VAR against the baseline VAR. Our method consistently corrects a wide range of structural and semantic errors present in the unguided baseline.

The quantitative results are summarized in Table 1. Our findings clearly demonstrate the efficacy of the AID-VAR framework. For each model size (d16, d20, d24), our guided approach consistently outperforms its unguided VAR counterpart, achieving a notable reduction in FID while maintaining or improving other metrics. For instance, AID-VAR-d16 improves the FID score from 3.55 to 3.24. This enhancement is achieved with a negligible increase in parameters (approx. 11M), underscoring the efficiency of the Guidance Injector. When compared to the broader landscape of generative models, AID-VAR demonstrates highly competitive performance against top-tier models like MAR and DiT-XL/2, validating our approach as a practical and effective method for post-hoc enhancement of large-scale generative models. Additional multi-seed and cross-backbone evaluations are provided in Appendices E.1 and E.3.

5.3 Correction and Guidance Analysis

To qualitatively assess the impact of our framework, Figure 5 compares the outputs of AID-VAR against the unguided baseline. The baseline is severely prone to structural and anatomical errors due to uncorrected error accumulation (*e.g.*, distorted eyes, erroneously produced "ghost wings"). In contrast, AID-VAR effectively rectifies these flaws, significantly enhancing global coherence and semantic plausibility.

To investigate the dynamics of these improvements, Figure 6 provides a step-by-step generative analysis, revealing two distinct phases: (1) **Early stages:** The trajectories of both models are visually indistinguishable, as early prediction errors are minuscule and have not yet caused structural failure. (2) **Mid-to-late stages:** A critical divergence emerges. Lacking correction, the baseline's minor errors compound into "structural hallucinations" (*e.g.*, rendering a malformed rooster). Conversely, AID-VAR preserves structural integrity because the Guidance Injector proactively mitigates these early deviations.

This visual analysis is quantitatively validated by the Inter-Scale Consistency Score (ISCS) in Figure 7. While both models exhibit comparable ISCS in the initial scales (0-4), a decisive divergence occurs during the high-resolution refinement phase (scales 7-9). AID-VAR's score surges dramatically (reaching 45.87 at scale 9), whereas the baseline stagnates due to its accumulated early errors.

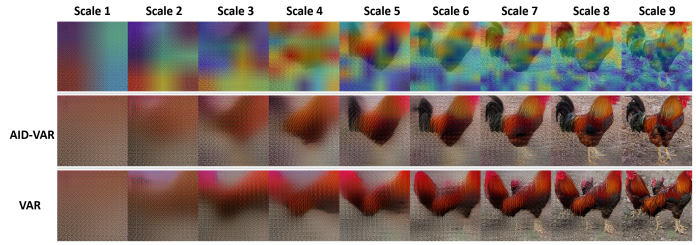


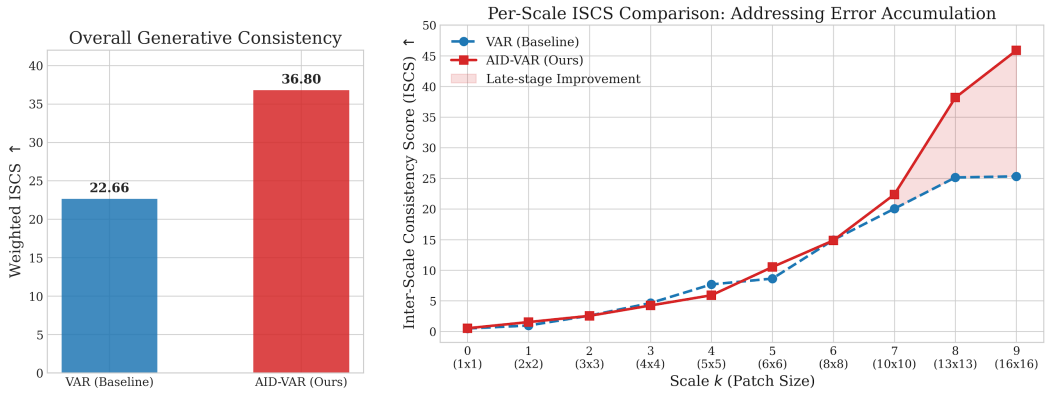
Figure 6: Visualization of AID-VAR as a Process Supervisor. This comparison shows three trajectories at each scale: (Top) The guidance features generated by our Guidance Injector, visualized as a heatmap; (Middle) The generative process of AID-VAR; (Bottom) The generative process of the baseline VAR.

This confirms our core argument: The Guidance Injector acts as a **"process supervisor"**. By neutralizing microscopic errors in early scales, it prevents catastrophic structural collapse, ultimately serving as an expert finisher that unlocks high-fidelity refinement in the final stages (see Appendices E.2, E.2, and B.3).

5.4 Ablation Study

We conduct a series of ablation studies to validate the key design choices of our AID-VAR framework. All ablations are performed on the VAR-d16 model.

Guidance method. We evaluate the importance of spatial information by comparing our spatially-aware guidance map with a single guidance vector broadcast across all locations. As shown in Table 2,



(a) Overall Consistency. (b) Per-scale Comparison.

Figure 7: Quantitative evaluation using the Inter-Scale Consistency Score (ISCS). (a) The weighted final scores demonstrate the overall generative consistency. (b) The per-scale analysis reveals the significant late-stage divergence, where AID-VAR effectively overcomes the error accumulation that plagues the baseline VAR.

the single-vector baseline only marginally improves FID from 3.55 to 3.43, whereas our spatial map reaches 3.24, demonstrating that spatially targeted corrections are crucial for resolving local structural errors. We further compare this dynamic spatial injection with static parameter-efficient fine-tuning (e.g., LoRA) in Appendix C.1.

Discriminator input space. We validate the necessity of a pixel-space discriminator by comparing it with one operating on VQ-VAE intermediate features. As shown in Table 2, feature-level training fails catastrophically (FID 4.17), indicating that pre-trained discriminators such as DINO struggle with abstract, quantized latents. This highlights the importance of continuous RGB inputs for providing meaningful adversarial signals; further analysis is provided in Appendices B.1, B.1, and B.2.

Table 2: Ablation studies on the ImageNet 256×256 validation set using VAR-d16 as the base model.

Experiment	Configuration	FID↓
Guidance method	Baseline (No Guidance)	3.55
	Single Token Broadcast	3.43
	Spatial Map (Ours)	**3.24**
Discriminator input	Baseline (No Guidance)	3.55
	VQ-VAE Features	4.17
	RGB Image (Ours)	**3.24**
Guidance weight (w)	Baseline (w=0)	3.55
	0.0001	3.60
	0.001 (Ours)	**3.24**
	0.01	4.32
	0.1	7.73

Guidance weight. Varying the inference guidance weight w yields a U-shaped performance curve (Table 2): too little guidance provides negligible correction, whereas excessive guidance ($w \geq 0.01$) disrupts the base VAR prior and degrades quality. We therefore use $w = 0.001$, which provides effective correction while preserving the frozen model's generative capability (see Appendix B.3 for dynamic guidance scheduling).

6 Conclusion

To address the detail loss and structural distortions in Visual Autoregressive (VAR) models caused by error accumulation, we propose AID-VAR, a lightweight, plug-and-play framework. Without altering the pre-trained and frozen VAR model, our method introduces a small guidance injector that corrects the generative process via Adversarially Injected Diagnosis. To ensure stable and efficient training, we introduce the key strategy: a differentiable path from an RGB-space discriminator to the injector, enabled by soft-label decoding. Furthermore, we propose the novel Inter-Scale Consistency Score (ISCS) to specifically quantify the fidelity of transitions between adjacent scales, directly measuring the mitigation of error accumulation. Experiments demonstrate that AID-VAR significantly improves the performance of VAR backbones (for instance, a 16% FID improvement for AID-VAR-d20) with a minimal parameter increase (only 3%) and computational overhead, establishing a practical and efficient pathway for upgrading existing large-scale generative models.

AI USE STATEMENT

We used AI-assisted tools to support manuscript editing, figure refinement, and LaTeX formatting. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the accuracy and integrity of the final manuscript.esponsible for the final text, claims, and artifacts.

ETHICS STATEMENT

Improving synthetic-image fidelity can facilitate misleading content and can preserve or amplify biases in pretrained models and training data. We advocate clear labeling of generated content and responsible deployment. Our evaluations concern ImageNet image synthesis and do not establish suitability for clinical or other safety-critical applications. Appendix F.2 discusses broader impacts and potential mitigations.

REPRODUCIBILITY STATEMENT

Section 3 describes the learning objective and inference procedure. Appendix D documents training settings, and Appendix E reports additional sampling-seed evaluations and controlled comparisons. Appendices A.1 and A.3 state the assumptions used in the conditional analysis. Details absent from the available experiment records are identified in Appendix F.1.

REFERENCES

Alpha-VLLM. Large-dit-imagenet. `https://github.com/Alpha-VLLM/LLaMA2-Accessory/tree/f7fe19834b23e38f333403b91bb0330afe19f79e/Large-DiT-ImageNet`, 2024.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

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*, pp. 11315–11325, 2022.

Cheng Cheng, Lin Song, Yicheng Xiao, Yuxin Chen, Xuchong Zhang, Hongbin Sun, and Ying Shan. Tensorar: Refinement is all you need in autoregressive image generation. *arXiv preprint arXiv:2505.16324*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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*, pp. 12873–12883, 2021.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James R Glass. Quantifying exposure bias for open-ended language generation. 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10124–10134, 2023.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11523–11532, 2022.

Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2142–2152, 2023.

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.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Jieru Lin, Danqing Huang, Tiejun Zhao, Dechen Zhan, and Chin-Yew Lin. Spot the error: Non-autoregressive graphic layout generation with wireframe locator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3413–3421, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Raghul Parthipan, Mohit Anand, Hannah M Christensen, J Scott Hosking, and Damon J Wischik. Defining error accumulation in ml atmospheric simulators. *arXiv preprint arXiv:2405.14714*, 2024.

Marco Pasini, Javier Nistal, Stefan Lattner, and George Fazekas. Continuous autoregressive models with noise augmentation avoid error accumulation. *arXiv preprint arXiv:2411.18447*, 2024.

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

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

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*, pp. 10684–10695, 2022.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *International conference on machine learning*, pp. 30105–30118. PMLR, 2023.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 157–167, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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.

Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pp. 292–309. Springer, 2024.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10459–10469, 2023.

Xiaoyu Yue, Zidong Wang, Yuqing Wang, Wenlong Zhang, Xihui Liu, Wanli Ouyang, Lei Bai, and Luping Zhou. Understand before you generate: Self-guided training for autoregressive image generation. *arXiv preprint arXiv:2509.15185*, 2025.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pp. 7354–7363. PMLR, 2019.

A Theoretical Analysis

A.1 Error Formulation and Accumulation

Let the VAR model contain K autoregressive scales, with corresponding resolutions $(p_1, p_2, \ldots, p_K)$ (e.g., $(1, 2, 3, \ldots, 16)$). At scale k, the model generates $n_k = p_k^2$ discrete tokens in parallel, each taking a value from the VQ codebook $\mathcal{V} = \{1, \ldots, V\}$. We define:

- $\mathbf{r}_k^* \in \mathcal{V}^{n_k}$: ground-truth token map at scale k encoded by the VQ-VAE;
- $\hat{\mathbf{r}}_k \in \mathcal{V}^{n_k}$: generated token map at scale k;
- $\mathbf{e} : \mathcal{V} \to \mathbb{R}^d$: VQ codebook embedding function;
- $\mathbf{h}_k^* = \mathrm{Reshape}(\mathbf{e}(\mathbf{r}_k^*))$ and $\hat{\mathbf{h}}_k = \mathrm{Reshape}(\mathbf{e}(\hat{\mathbf{r}}_k))$: feature maps in $\mathbb{R}^{d \times p_k \times p_k}$;
- $\Phi_k : \mathbb{R}^{d \times p_k \times p_k} \to \mathbb{R}^{d \times H \times H}$: the upsampling and residual-convolution operator, where $H = p_K$.

The accumulated feature maps of the reference and generated trajectories are

$$f_k^* = \sum_{j=1}^{k} \Phi_j(\mathbf{h}_j^*), \qquad \hat{f}_k = \sum_{j=1}^{k} \Phi_j(\hat{\mathbf{h}}_j).$$

The autoregressive process for $k = 1, \ldots, K$ is

$$\mathbf{x}_k = \mathrm{AreaDownsample}(\hat{f}_{k-1}, p_k), \qquad \hat{\mathbf{r}}_k \sim \mathrm{Transformer}(\mathbf{x}_k, c), \tag{11}$$

with the initial state $\hat{f}_0 = \mathbf{0}$.

Scale and cumulative error. The feature error introduced at scale k is

$$\boldsymbol{\delta}_k \Phi_k(\hat{\mathbf{h}}_k) - \Phi_k(\mathbf{h}_k^*),$$

and the cumulative error after k scales is

$$\boldsymbol{\Delta}_k \hat{f}_k - f_k^* = \sum_{j=1}^{k} \boldsymbol{\delta}_j.$$

Proposition 1 (Additive decomposition). The final cumulative error is the sum of the individual scale errors, $\boldsymbol{\Delta}_K = \sum_{k=1}^{K} \boldsymbol{\delta}_k$, and

$$\mathbb{E}\left[\|\boldsymbol{\Delta}_K\|^2\right] = \sum_{k=1}^{K} \mathbb{E}\left[\|\boldsymbol{\delta}_k\|^2\right] + 2 \sum_{1 \le i < j \le K} \mathbb{E}\left[\langle \boldsymbol{\delta}_i, \boldsymbol{\delta}_j \rangle\right]. \tag{12}$$

Intrinsic sampling error. **Proposition 2 (Conditional lower bound).** Suppose distinct codebook vectors have minimum distance $d_{\min} > 0$, at least one token has a nonzero error probability $\eta_k^{(i)}$, and Φ_k has a lower gain $m_k > 0$ on the relevant embedding differences:

$$\|\Phi_k(u) - \Phi_k(v)\| \ge m_k \|u - v\|.$$

Then, for a ground-truth input context,

$$\mathbb{E}\left[\|\boldsymbol{\delta}_k\|^2\right] \ge m_k^2 d_{\min}^2 \sum_i \eta_k^{(i)} =: v_k > 0. \tag{13}$$

A mistaken token contributes at least $d_{\min}^2$ to the squared embedding error. Summing the expectations over positions and applying the assumed lower gain proves the result. An operator norm supplies an upper bound and cannot replace m_k in this argument. If the mapping has a null space on the relevant differences, or predictions are deterministic and correct, strict positivity is not guaranteed.

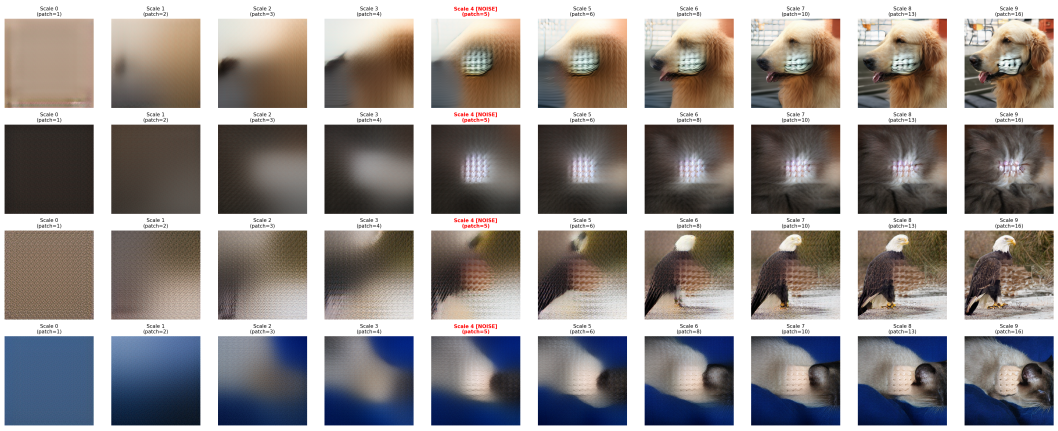


Figure 8: A noise-injection example of autoregressive error propagation. We intentionally inject noise patches at Scale 4 during generation. The standard VAR model, lacking an active correction mechanism, fails to identify the injected noise as an artifact and propagates the distortion through subsequent scales (Scales 5–9). This finite example illustrates persistent deviation rather than establishing a universal propagation bound.

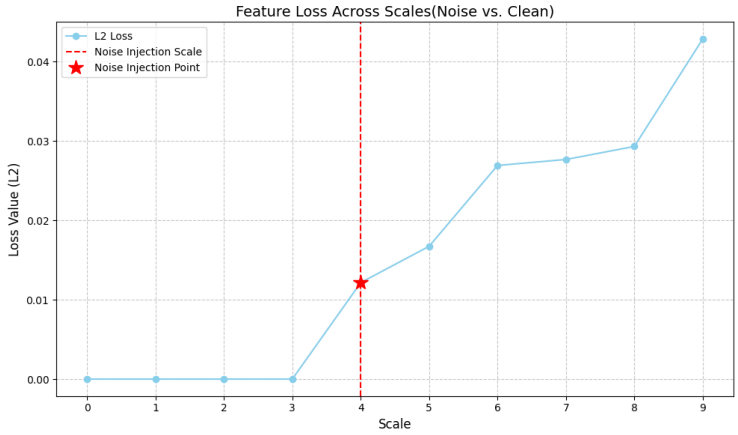


Figure 9: Cross-scale cumulative feature loss ($\|\boldsymbol{\Delta}_k\|^2$) after noise injection. The feature difference between the clean and perturbed trajectories increases after the injection point at Scale 4. The observed finite trajectory illustrates amplification in this experiment but does not establish an asymptotic growth rate.

A.2 Cross-Scale Error Amplification

Assumption 1 (Lipschitz condition). The model satisfies

$$W_2\left(p_\theta(\mathbf{h}_k \mid f, c), p_\theta(\mathbf{h}_k \mid f', c)\right) \leq L_k \|f - f'\|. \tag{14}$$

Assumption 1 is an upper-sensitivity condition and alone does not imply error amplification. For the following lower-bound analysis, we additionally assume

$$\boldsymbol{\delta}_k = \boldsymbol{\epsilon}_k + \boldsymbol{\pi}_k,$$

where

$$\mathbb{E}\|\boldsymbol{\epsilon}_k\|^2 \geq \sigma_k^2, \qquad \mathbb{E}\langle \boldsymbol{\epsilon}_k, \boldsymbol{\pi}_k \rangle = 0,$$

and

$$\mathbb{E}\|\boldsymbol{\pi}_k\|^2 \geq a_k^2 \mathbb{E}\|\boldsymbol{\Delta}_{k-1}\|^2.$$

Here a_k is an assumed amplification coefficient and should not be identified with the Lipschitz upper bound L_k.

Theorem 1 (Conditional scale-error bound). Under the assumptions above,

$$\mathbb{E}\left[\|\boldsymbol{\delta}_k\|^2\right] \geq \sigma_k^2 + a_k^2 \mathbb{E}\left[\|\boldsymbol{\Delta}_{k-1}\|^2\right]. \tag{15}$$

Figures 8 and 9 motivate this amplification model but do not verify all of its moment assumptions.

Theorem 2 (Conditional cumulative growth). In addition to Theorem 1, assume

$$\mathbb{E}\langle \boldsymbol{\Delta}_{k-1}, \boldsymbol{\delta}_k \rangle \geq 0, \qquad a_k \geq a > 0, \qquad \sigma_k^2 \geq \sigma^2 > 0.$$

Starting from zero cumulative error,

$$\mathbb{E}\left[\|\boldsymbol{\Delta}_K\|^2\right] \geq \sigma^2 \frac{(1+a^2)^K - 1}{a^2}. \tag{16}$$

The assumptions give

$$\mathbb{E}\left[\|\boldsymbol{\Delta}_k\|^2\right] \geq (1+a^2)\mathbb{E}\left[\|\boldsymbol{\Delta}_{k-1}\|^2\right] + \sigma^2.$$

Unrolling the recurrence from $\mathbb{E}\|\boldsymbol{\Delta}_1\|^2 \geq \sigma^2$ gives the geometric series.

Under the additional moment assumptions, this lower bound grows at the rate of $(1+a^2)^K$.

Spatial amplification. Let $\mathcal{U}_k$ denote the upsampling operator. We write

$$\|\mathcal{U}_k(\boldsymbol{\delta})\|^2 = \left(\frac{H}{p_k}\right)^2 \|\boldsymbol{\delta}\|^2 \gamma_k, \tag{17}$$

where γ_k records the operator-dependent scaling. For nearest-neighbor replication with an integer scale ratio, $\gamma_k = 1$, whereas general interpolation or convolution operators need not have this value.

Table 3: Comparison of error accumulation mechanics.

Metric	Standard AR	VAR
Steps	N (e.g., 256)	K (e.g., 10)
Propagation	$O((1+a^2)^N)$	$O((1+a^2)^K)$
Spatial Amplification	None	Significant (Upsampling)

Overall, error accumulation can arise from token errors, correlated cross-scale propagation, and spatial upsampling. The geometric lower bound above is conditional on the stated amplification assumptions; neither discrete tokenization nor Lipschitz continuity alone implies this regime.

A.3 Error Containment with AID-VAR

Appendix A.1 characterizes how scale-wise errors can accumulate under a conditional amplification model. We next derive an upper bound under an additional negative-feedback hypothesis for AID-VAR. The adversarial objective itself does not guarantee this hypothesis, and the representation measurements in Appendix E.2 provide indirect empirical evidence rather than a proof.

In standard VAR, the scale error is decomposed as

$$\boldsymbol{\delta}_k = \boldsymbol{\epsilon}_k + \boldsymbol{\pi}_k,$$

where $\boldsymbol{\epsilon}_k$ denotes intrinsic sampling error and $\boldsymbol{\pi}_k$ denotes propagation error.

In AID-VAR, the guidance injector P_ϕ processes the historical feature map and produces a spatial guidance signal $\mathbf{G}_k$. The hidden input is modified as described in Section 3,

$$z'_k = f_{\mathrm{VAR}}\left(x_k + w\mathbf{G}_k \mid \{S_{<k}\}\right),$$

resulting in a guided feature map $\tilde{\mathbf{h}}_k$. We define the guided scale error as

$$\tilde{\boldsymbol{\delta}}_k \Phi_k(\tilde{\mathbf{h}}_k) - \Phi_k(\mathbf{h}_k^*) = \boldsymbol{\epsilon}_k + \boldsymbol{\pi}_k - \boldsymbol{c}_k, \tag{18}$$

where $\boldsymbol{c}_k$ is the active correction introduced by the guidance injector.

The RGB discriminator supplies a perceptual training signal. A useful correction would oppose the propagated deviation, schematically $\boldsymbol{c}_k \approx \gamma_k \boldsymbol{\pi}_k$. This is a modeling hypothesis: hinge-loss adversarial training does not by itself imply this equality, Wasserstein-distance minimization, or the negative-feedback condition introduced below.

Assumption 2 (Adversarial negative feedback). Assume that the residual propagation error is nonpositively correlated with the historical deviation:

$$\mathbb{E}\left[\left\langle \tilde{\boldsymbol{\Delta}}_{k-1}, \boldsymbol{\pi}_k - \boldsymbol{c}_k \right\rangle\right] \leq 0. \tag{19}$$

This signed-moment condition is an explicit assumption rather than a consequence of a well-trained discriminator. We further assume zero-mean sampling noise conditional on the past, orthogonality between sampling noise and residual propagation, and an upper variance bound

$$\mathbb{E}\|\boldsymbol{\epsilon}_k\|^2 \leq \sigma_k^2.$$

Here σ_k^2 is an upper bound and is distinct from the lower bound used in the amplification analysis above.

Theorem 3 (AID-VAR error containment). Under Assumptions 1 and 2, suppose the effective propagation error is damped by $\gamma \in (0, 1]$ such that

$$\mathbb{E}\left[\|\boldsymbol{\pi}_k - \boldsymbol{c}_k\|^2\right] \leq (1-\gamma)^2 L^2 \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right].$$

Then

$$\mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_k\|^2\right] \leq \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right] + \sigma_k^2 + (1-\gamma)^2 L^2 \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right]. \tag{20}$$

Proof. By definition,

$$\tilde{\boldsymbol{\Delta}}_k = \tilde{\boldsymbol{\Delta}}_{k-1} + \tilde{\boldsymbol{\delta}}_k.$$

Expanding the squared norm gives

$$\begin{aligned} \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_k\|^2\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right] + \mathbb{E}\left[\|\tilde{\boldsymbol{\delta}}_k\|^2\right] \\ + 2\mathbb{E}\left[\left\langle \tilde{\boldsymbol{\Delta}}_{k-1}, \tilde{\boldsymbol{\delta}}_k \right\rangle\right]. \end{aligned} \tag{21}$$

Substituting $\tilde{\boldsymbol{\delta}}_k = \boldsymbol{\epsilon}_k + (\boldsymbol{\pi}_k - \boldsymbol{c}_k)$, the conditional zero-mean assumption gives

$$\mathbb{E}\left[\left\langle \tilde{\boldsymbol{\Delta}}_{k-1}, \boldsymbol{\epsilon}_k \right\rangle\right] = 0.$$

Using Assumption 2,

$$2\mathbb{E}\left[\left\langle \tilde{\boldsymbol{\Delta}}_{k-1}, \tilde{\boldsymbol{\delta}}_k \right\rangle\right] \leq 0. \tag{22}$$

Under the orthogonality assumption,

$$\begin{aligned} \mathbb{E}\left[\|\tilde{\boldsymbol{\delta}}_k\|^2\right] &= \mathbb{E}\left[\|\boldsymbol{\epsilon}_k\|^2\right] + \mathbb{E}\left[\|\boldsymbol{\pi}_k - \boldsymbol{c}_k\|^2\right] \\ &\leq \sigma_k^2 + (1-\gamma)^2 L^2 \mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right]. \end{aligned} \tag{23}$$

Combining these bounds proves the result.

Conditional cumulative upper bound. Let

$$\tilde{L} = (1-\gamma)L$$

be the effective propagation constant under adversarial guidance. If $\sigma_k^2 \leq \sigma^2$ for all scales, Theorem 3 gives

$$\mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_k\|^2\right] \leq (1+\tilde{L}^2)\mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{k-1}\|^2\right] + \sigma^2. \tag{24}$$

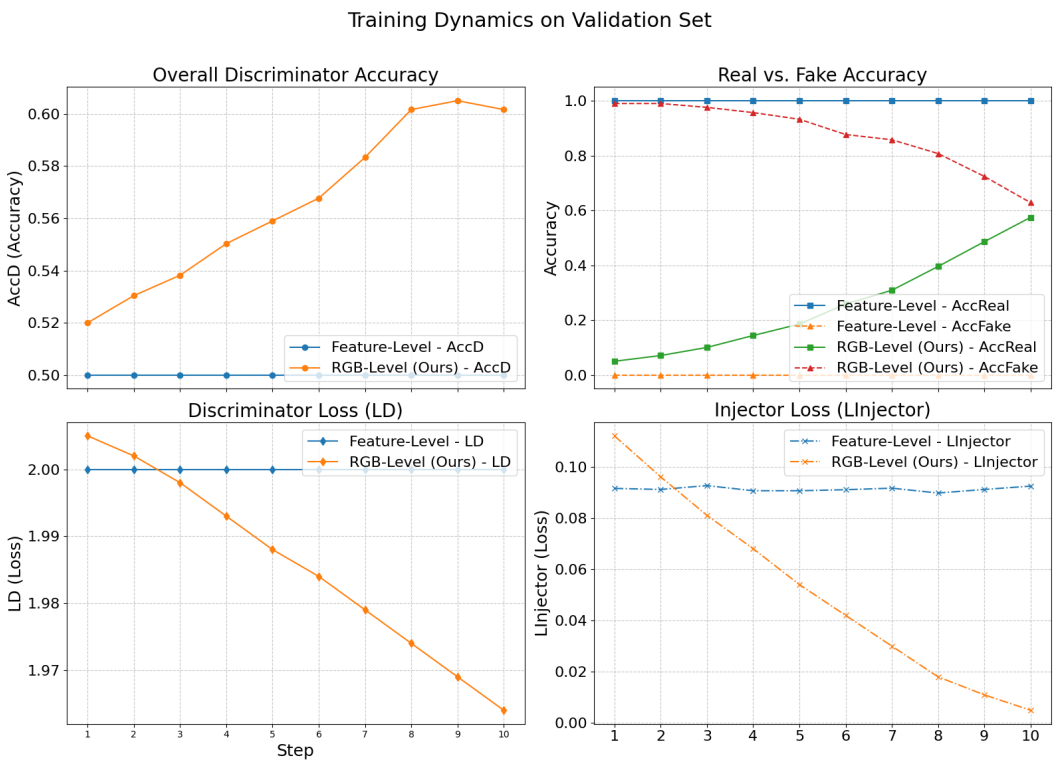


Figure 10: Training dynamics for different discriminator input spaces. The original VQ-VAE feature-level discriminator (Blue/Dashed) collapses to a trivial solution, with Real Accuracy remaining near 100%, Fake Accuracy near 0%, and discriminator loss near 2.0. In contrast, RGB-level discrimination through soft decoding exhibits substantially healthier adversarial optimization and a decreasing injector loss.

Unrolling across K scales yields

$$\mathbb{E}\left[\|\tilde{\mathbf{\Delta}}_K\|^2\right] \leq \sigma^2 \frac{(1+\tilde{L}^2)^K - 1}{\tilde{L}^2}. \tag{25}$$

For $\gamma = 1$, the zero-denominator expression is interpreted by continuity and becomes $K\sigma^2$. For $0 < \gamma < 1$, the bound remains geometric; it does not imply uniformly bounded error or asymptotically linear growth. Furthermore, comparing upper and lower bounds obtained under different assumptions does not by itself establish a measured improvement unless those assumptions are verified.

B Analysis of Adversarial Guidance

B.1 Discriminator Input Space

A critical design choice in AID-VAR is the representation space in which the adversarial objective is applied. We first compare two strategies: (1) discrimination using VQ-VAE-level latent features and (2) discrimination in RGB space through differentiable soft decoding.

Original VQ-VAE feature discriminator. The original feature-level baseline extracts features using the VQ-VAE encoder and feeds them directly to a discriminator head. In our experiments, this configuration exhibits a rapid collapse. As shown in Figure 10, the discriminator eventually classifies nearly all samples as "Real": *AccReal* remains near 100% while *AccFake* approaches 0%. The discriminator loss consequently stabilizes near 2.0, while the injector receives little useful adversarial signal.

RGB-space discrimination. Our soft-decoding strategy instead projects the token distribution into RGB space before discrimination. This provides two practical advantages:

- **Image-space projection.** Decoding transforms abstract token representations into structured visual signals where perceptual artifacts are easier to identify.
- **Pre-trained visual priors.** RGB images can be processed by the frozen DINO backbone, providing semantically meaningful features for the discriminator.

The RGB-level curves in Figure 10 show a substantially healthier adversarial game. The real and fake accuracies evolve toward a more balanced regime, the discriminator loss decreases from approximately 2.0 to 1.64, and the injector loss decreases from approximately 0.11 to 0.005.

Stronger native-feature discriminator. The failure of the original VQ-VAE-feature baseline does not imply that feature-space adversarial supervision is inherently ineffective. We therefore additionally use the frozen pretrained VAR backbone as a native feature extractor for a LADD-style discriminator (VAR-D). This stronger baseline avoids the collapse observed with the original VQ-VAE feature discriminator.

Table 4: Discriminator input-space comparisons on VAR-d16. The original VQ-VAE-feature result is included for context.

Method	FID ↓
Unguided VAR-d16	3.55
Original VQ-VAE-feature discriminator	4.17
Native VAR-feature discriminator (VAR-D)	3.37
RGB discriminator with soft decoding (AID-VAR)	**3.24**

The VAR-D result shows that feature-level adversarial supervision can be effective with an appropriate representation extractor. RGB soft decoding performs better in this comparison, although these FID values do not establish a universal ordering between input spaces or directly quantify gradient smoothness. Here "LADD-style" refers to the use of a pretrained generator feature space, following Sauer et al., *Fast High-Resolution Image Synthesis with Latent Adversarial Diffusion Distillation*, SIGGRAPH Asia 2024.

B.2 Differentiable Decoding

Soft decoding provides a differentiable path from image-space supervision back to the guidance injector. It is an empirical design choice rather than a sufficient condition for the negative-feedback assumption used in Appendix A.3. The successful VAR-D experiment above also shows that RGB-space discrimination is not strictly necessary for adversarial correction.

To evaluate the decoding choice, we compare soft-label decoding with the straight-through estimator (STE), which performs hard token selection in the forward pass while using a surrogate gradient in the backward pass.

Table 5: Gradient-estimator comparison on VAR-d16. No numerical FID is available for the unstable Gumbel-Softmax runs.

Configuration	FID ↓
Unguided VAR-d16	3.55
Guidance trained with STE	3.35
Guidance trained with soft-label decoding	**3.24**
Guidance trained with Gumbel-Softmax	Unstable

The completed STE experiment achieves an FID of 3.35, compared with 3.24 for soft-label decoding and 3.55 for the unguided baseline.

Qualitative Gumbel-Softmax trials are sensitive to temperature. Large temperatures produce overly soft distributions and plateauing discriminator losses, whereas small temperatures produce nearly one-hot distributions and ineffective injector updates. Simple temperature annealing across scales did not yield reliably stable training in the tested configurations. These observations remain qualitative because temperatures and run-level statistics were not retained in the available summary.

Soft decoding directly differentiates through the continuous surrogate and therefore avoids the forward/backward mismatch of STE. However, it is not an unbiased estimator of the original hard-token objective, and a soft-training/hard-inference gap remains. The observed FID comparison supports this design in the tested setting without establishing general optimality.

B.3 Guidance Placement and Scheduling

Intermediate-scale versus terminal-only guidance. To distinguish cross-scale correction from generic final-stage adversarial refinement, we restrict guidance to the final generation scale (Scale 10). Terminal-only refinement improves FID from 3.55 to 3.41, whereas cross-scale injection reaches 3.24.

Table 6: Placement of guidance on VAR-d16.

Method	FID ↓
Unguided VAR-d16	3.55
Terminal refinement (Scale 10 only)	3.41
AID-VAR (cross-scale injection)	**3.24**

The additional 0.17 FID reduction obtained by cross-scale injection supports the contribution of intervention before the final scale. This result does not imply that all possible terminal refinement methods would behave similarly.

Coarse-to-fine guidance scheduling. We additionally evaluate a linear decay schedule that applies stronger guidance at coarse scales and weaker guidance at fine scales.

Table 7: Static and dynamically decaying guidance weights.

Guidance configuration	FID ↓
No guidance	3.55
Static weight ($w = 0.001$)	**3.24**
Linear decay across scales	3.28

Linear decay improves FID by 0.27 over the unguided baseline and trails the selected static guidance weight by 0.04. This provides evidence for an alternative scale-dependent policy but does not establish a wider stable range or eliminate the need for endpoint selection. For new datasets, monitoring $\mathcal{L}_{\text{rec}}$ while calibrating the guidance strength provides a practical heuristic: weights that destabilize reconstruction should be avoided even if the adversarial objective improves.

C Comparison with Alternative Adaptation

C.1 LoRA Fine-Tuning Baseline

A natural alternative to an external guidance injector is parameter-efficient fine-tuning. We therefore consider Low-Rank Adaptation (LoRA) modules inserted into the self-attention and feed-forward layers of the frozen VAR backbone and optimized using the same adversarial objective.

Optimization behavior. In AID-VAR, the guidance injector P_ϕ is a shallow external module attached to the interface of the target scale. The adversarial gradient therefore propagates through a comparatively short trainable path.

By contrast, updating internal LoRA parameters requires the image-space adversarial signal to propagate through the deep VAR computational graph before reaching the inserted low-rank matrices. In our tested configurations, this optimization exhibited unstable training behavior, including late-stage collapse and invalid inference checkpoints.

These observations characterize the tested configurations rather than establishing that adversarial LoRA fine-tuning is intrinsically unstable.

Difference from trajectory-conditioned correction. LoRA modifies parameters that are shared across generation scales, whereas AID-VAR predicts an explicit correction conditioned on the current generation trajectory while preserving the original backbone parameters. This difference motivates the comparison, but it does not imply that weight adaptation cannot reduce cross-scale errors.

C.2 Training Overhead and Scalability

Although LoRA introduces fewer trainable parameters, it is not necessarily memory-efficient for this adversarial training setup. Gradients must propagate to internal LoRA modules, requiring storage of intermediate activations throughout the frozen backbone. In contrast, AID-VAR treats the VAR backbone as an inference engine and restricts the trainable computational graph to the lightweight guidance injector, differentiable decoder path, and discriminator.

All measurements below were collected on a single compute node with 8 NVIDIA A800 (80GB) GPUs and a global batch size of 256.

Table 8: Quantitative comparison of training overhead across VAR backbone scales. Peak GPU memory is reported per device.

Method	Trainable Params	Frozen Params	Total Params	Peak GPU Memory	Training Time
Base Model: VAR-d16 (310M)					
LoRA ($r = 16$)	4.5 M	310 M	314.5 M	∼46.2 GB	∼15.5 hours
AID-VAR (Ours)	**11.0 M**	310 M	321.0 M	**∼14.8 GB**	**∼9.2 hours**
Base Model: VAR-d20 (600M)					
LoRA ($r = 16$)	8.2 M	600 M	608.2 M	∼71.5 GB	∼22.0 hours
AID-VAR (Ours)	**19.0 M**	600 M	619.0 M	**∼18.2 GB**	**∼10.5 hours**
Base Model: VAR-d24 (1.0B)					
LoRA ($r = 16$)	12.8 M	1.0 B	≈1.01 B	**OOM** (>80 GB)	**N/A**
AID-VAR (Ours)	**27.0 M**	1.0 B	≈1.02 B	**∼24.5 GB**	**∼11.8 hours**

For VAR-d16, LoRA uses fewer trainable parameters (4.5M versus 11.0M) but requires substantially more training memory. For the VAR-d24 configuration, activation storage leads to an OOM failure at the tested batch size on an 80GB device.

We additionally test the resource overhead of LoRA with the stronger VAR-D discriminator introduced in Appendix B.1.

Table 9: Training overhead on VAR-d16 (310M frozen parameters). Peak memory and wall-clock training time are approximate.

Method	Trainable	Frozen	Peak memory	Time
LoRA (DINO-D)	4.5M	310M	46.2 GB	15.5 h
LoRA (VAR-D)	4.5M	310M	45.8 GB	14.8 h
AID-VAR	11.0M	310M	14.8 GB	9.2 h

Using VAR-D does not remove the activation-storage cost of updating internal LoRA parameters. The measured peak memory remains approximately $3.1\times$ that of AID-VAR. A valid generation-quality result for the strengthened LoRA (VAR-D) configuration is not available in the supplied experiments, so this comparison is restricted to resource consumption. Other possible variants, such as combined R1/R2 regularization, scratch discriminators, or approximated gradient penalties, were not accompanied by completed results and are therefore not reported as evaluated baselines.

Inference complexity. AID-VAR introduces only a lightweight guidance injector operating on features already produced during VAR generation. Guidance is incorporated through a simple additive modification while the original VAR backbone remains unchanged. The additional parameter count is approximately 11M, 19M, and 27M for VAR-d16, VAR-d20, and VAR-d24, respectively, corresponding to only a small fraction of the underlying backbone size. This design therefore preserves the plug-and-play nature of the pretrained generator while avoiding full-backbone fine-tuning.

D Implementation Details

D.1 Model Architectures

Guidance injector. The Guidance Injector is a lightweight Transformer encoder with two Transformer blocks and eight attention heads per block. Its embedding dimension (`PLANNER_DIM`) is matched to the VAR model depth:

$$\texttt{PLANNER_DIM} = \texttt{VAR_DEPTH} \times 64,$$

which gives, for example, 1536 dimensions for VAR-d24.

Within each block, the feed-forward network uses hidden dimension

$$\min(\texttt{embed_dim} \times 2, 512)$$

with GELU activation. This corresponds to approximate MLP ratios of 0.5, 0.4, and 0.33 for VAR-d16, VAR-d20, and VAR-d24, respectively. We additionally employ a custom `SafeLayerNorm`, scale residual connections by 0.1 for training stability, and add sinusoidal positional encodings to the input features.

Discriminator. Our discriminator follows the Projected Discriminator design. We use a pretrained DINO ViT-S/16 model (Caron et al., 2021) as a frozen visual feature backbone. Only the shallow discriminator heads are trainable.

Each head contains a main block followed by a classification layer. The main block uses a 1×1 convolution followed by a residual block containing a 9×9 convolution. The classification layer is implemented as a 1×1 `SpectralConv1d` layer. The DINO backbone remains frozen throughout training.

D.2 Training Setup and Hyperparameters

All experiments were conducted on a server equipped with 8 NVIDIA A800 GPUs. Training the Guidance Injector and discriminator for each VAR model variant on the full ImageNet dataset took approximately 9 hours. Our implementation uses PyTorch.

We optimize both the Guidance Injector and discriminator heads using AdamW with

$$\beta_1 = 0.9, \qquad \beta_2 = 0.999, \qquad \text{weight decay} = 1 \times 10^{-5}.$$

A constant learning rate is used throughout training, as we did not observe a significant benefit from learning-rate warmup or decay in our setting.

E Additional Experimental Results

Unless otherwise stated, the controlled comparisons in this section use VAR-d16 and ImageNet at 256×256 resolution. Baseline and guided evaluations use the same CFG and token-sampling settings.

E.1 Robustness Across Sampling Seeds

We evaluate each fixed checkpoint using three random sampling seeds and 50,000 generated images per seed. All models use ten generation scales. Table 11 reports the mean and standard deviation across the three evaluations.

Table 10: Key hyperparameters for full ImageNet training and single-class training.

Parameter	**Full ImageNet**	**Single-class**
VAR Depth	16	16
Epochs	2	10
Global Batch Size	256	32
LR (Guidance Injector)	1×10^{-6}	5×10^{-6}
LR (Discriminator)	1×10^{-6}	1×10^{-6}
Gradient Clip Value	0.5	–
λ_{rec}	0.01	0.01
Guidance Weight (train)	0.001	0.001
R1 Gamma	0.2	–

Table 11: Mean $\pm$ standard deviation over three sampling seeds, with 50,000 generated images per seed. Checkpoints remain fixed across evaluations.

Model	Params	FID ↓	IS ↑	Precision ↑	Recall ↑
VAR-d16	310M	3.55 ± 0.07	274.4 ± 0.9	0.84 ± 0.006	0.51 ± 0.008
AID-VAR-d16	321M	3.24 ± 0.06	280.0 ± 1.1	0.85 ± 0.007	0.51 ± 0.009
VAR-d20	600M	2.95 ± 0.05	302.6 ± 1.3	0.83 ± 0.008	0.56 ± 0.007
AID-VAR-d20	619M	2.54 ± 0.08	309.4 ± 1.0	0.83 ± 0.006	0.56 ± 0.009
VAR-d24	1.0B	2.33 ± 0.06	312.9 ± 1.2	0.82 ± 0.007	0.59 ± 0.010
AID-VAR-d24	1.02B	2.08 ± 0.05	316.3 ± 1.5	0.82 ± 0.009	0.60 ± 0.008

The FID reductions are 0.31, 0.41, and 0.25 for d16, d20, and d24, respectively, corresponding to approximately 8.7%, 13.9%, and 10.7% relative improvements over their baselines. Each reduction exceeds the reported standard deviation of the corresponding fixed-checkpoint evaluations. These results support robustness to sampling randomness but do not measure training-run variability or constitute a formal significance test.

E.2 Cross-Scale Correction Analysis

Perceptual and structural consistency. ISCS and the projected discriminator both rely on DINO features. To provide complementary evidence, we additionally evaluate intermediate outputs against ground-truth references using LPIPS and SSIM.

Table 12: Intermediate-scale perceptual and structural measurements against GT references. These measurements complement rather than redefine cross-scale consistency.

Scale	LPIPS (VAR) ↓	LPIPS (AID) ↓	SSIM (VAR) ↑	SSIM (AID) ↑
1	0.675	0.688	0.324	0.311
3	0.546	0.531	0.463	0.478
5	0.421	0.408	0.597	0.614
7	0.317	0.258	0.685	0.749
9	0.294	0.187	0.701	0.822
10	0.305	0.164	0.688	0.847

AID-VAR is slightly worse at Scale 1. At later scales, however, its LPIPS decreases while SSIM increases, whereas VAR degrades between Scales 9 and 10. This pattern is consistent with improved late-stage structure under the reported reference-based evaluation.

For class-conditional generation, however, a unique paired GT image is not implicit in the task. Interpretation and exact replication therefore require the reference-pairing protocol described in Appendix F.1. These measurements alone do not establish trajectory consistency or rule out effects caused by the pairing procedure.

Intermediate-to-final representation alignment. We further compute Centered Kernel Alignment (CKA) between intermediate-scale representations and the final-scale representation within each model's generation trajectory. The final scale serves as the reference representation.

Table 13: CKA similarity between intermediate- and final-scale representations.

Intermediate scale	VAR	AID-VAR
1	0.124	0.108
2	0.135	0.152
3	0.201	0.178
4	0.245	0.268
5	0.358	0.321
6	0.442	0.575
7	0.518	0.751
8	0.582	0.869
9	0.691	0.945

AID-VAR exhibits lower alignment at Scales 1, 3, and 5 but substantially higher alignment at Scales 6–9. At Scale 9, CKA increases from 0.691 to 0.945.

Together with the terminal-only ablation and independent perceptual metrics, this pattern supports the proposed cross-scale interpretation. CKA nevertheless remains an association measure: it does not directly measure the signed correction term in Assumption 2, prove adversarial negative feedback, or establish causality. The theoretical containment result therefore remains conditional on its stated assumptions.

E.3 Transfer to FlexVAR

To examine transfer beyond the original VAR backbone, we attach the same guidance mechanism to a frozen FlexVAR model, an additional next-scale autoregressive architecture that removes residual prediction (Jiao et al., *FlexVAR: Flexible Visual Autoregressive Modeling without Residual Prediction*).

Table 14: Additional backbone evaluation using FlexVAR.

Method	FID ↓
FlexVAR	3.05
AID-FlexVAR	**2.81**

The FlexVAR baseline reaches an FID of 3.05, while AID-FlexVAR reaches 2.81, corresponding to an absolute reduction of 0.24 (approximately 7.9%). This result supports transfer within the tested next-scale generation family, although broader architectural generalization remains unverified.

F Limitations and Broader Impact

F.1 Limitations and Reporting Scope

Several limitations remain. First, our main experiments use a manually selected static guidance weight. A linear decay schedule also improves FID (Appendix B.3), but broader sensitivity analysis and cross-dataset calibration remain unverified.

Second, FlexVAR provides an additional backbone evaluation (Appendix E.3), but experiments on other datasets, generation paradigms, and higher resolutions remain future work.

Third, although AID-VAR mitigates error accumulation in autoregressive image generation, its behavior has not yet been validated for autoregressive video generation. In that setting, errors can accumulate both spatially and temporally, potentially leading to semantic drift or temporal

inconsistency over long sequences. The computational overhead may also increase substantially for long-form video generation.

Reporting limitations. The available experiment summaries omit several implementation details needed for exact replication, including: sampling-seed identifiers; the LPIPS encoder choice; GT-pairing and resizing protocols; the CKA layer, kernel, spatial-alignment procedure, and sample count; the exact FlexVAR checkpoint/configuration; and the endpoints of the guidance schedule. These quantities are therefore not inferred here. Among the reported evaluations, only Table 11 includes explicit sampling-seed dispersion estimates.

F.2 Broader Impact

Positive impacts. AID-VAR provides a lightweight mechanism for improving existing large-scale visual autoregressive generators without retraining the entire backbone. Such improvements may benefit applications including visual content creation, concept design, and image reconstruction. More broadly, the framework explores how post-hoc guidance can detect and correct deviations that emerge during a pretrained model's generation trajectory.

Potential risks. As with other generative-model improvements, higher image quality can also increase the realism of synthetic media and may therefore facilitate misuse, including deceptive or misleading image generation. Adversarial training may additionally inherit or amplify biases already present in the underlying training data. Although AID-VAR is lightweight relative to full-model training, the additional optimization still introduces computational and energy costs.

Mitigation strategies. Potential safeguards include synthetic-content labeling, watermarking, and complementary detection systems. Bias-related risks should be addressed through careful dataset selection and systematic evaluation across relevant demographic or semantic groups. Finally, more efficient training and optimization methods can further reduce the environmental cost of adapting large generative models.