FARMER: Flow AutoRegressive Transformer over Pixels

Guangting Zheng[1,3], Qinyu Zhao[2,3], Tao Yang[3], Fei Xiao[3], Zhijie Lin[4],
Jie Wu[3], Jiajun Deng[1], Yanyong Zhang[1✉], Rui Zhu[3]

[1]University of Science and Technology of China [2]Australian National University [3]ByteDance Seed China [4]ByteDance Seed Singapore

zgt@mail.ustc.edu.com, {dengjj,yanyongz}@ustc.edu.cn, zhurui.kim@bytedance.com

Abstract

Directly modeling the explicit likelihood of the raw data distribution is a key topic in the machine learning area, which achieves the scaling success in Large Language Models by autoregressive modeling. However, continuous AR modeling over visual pixel data suffers from extremely long sequences and high-dimensional spaces. In this paper, we present FARMER, a novel end-to-end generative framework that unifies Normalizing Flows (NFs) and Autoregressive (AR) models for tractable likelihood estimation and high-quality image synthesis directly from raw pixels. FARMER employs an invertible autoregressive flow to transform images into latent sequences, whose distribution is implicitly modeled by an autoregressive model. To address the redundancy and complexity in pixel-level modeling, we propose a self-supervised dimension reduction scheme that partitions NF latent channels into informative and redundant groups, enabling more effective and efficient AR modeling. Furthermore, we design a one-step distillation scheme to significantly accelerate inference speed and introduce a resampling-based classifier-free guidance algorithm to boost image generation quality. Extensive experiments demonstrate that FARMER achieves competitive performance compared to existing pixel-based generative models while providing exact likelihoods and scalable training.

1. Introduction

Explicitly modeling a normalized likelihood $\mathbf{P}(x)$ over the high-dimensional data distribution is challenging. Popular generative paradigms such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion/score-based models do not provide tractable likelihoods—VAEs optimize a lower bound, GANs learn implicit generators without likelihoods, and diffusion/score-based models offer likelihoods only via variational bounds or costly numerical estimation by a probability-flow ODE.

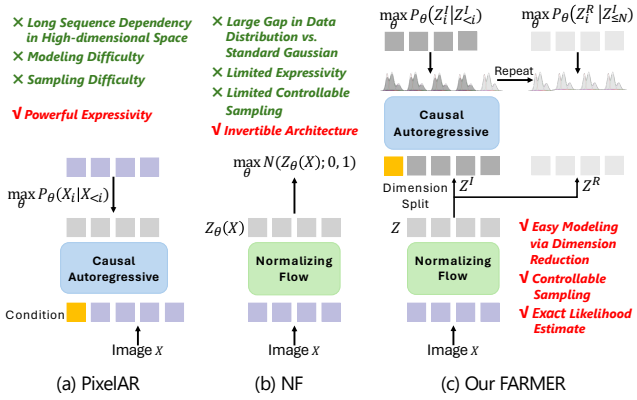


Figure 1. Autoregressive (AR) models offer strong expressivity but struggle with pixel modeling and sampling due to long sequences of high-resolution images. Normalizing flows (NFs) employ invertible mappings to transform complex image distributions to a standard Gaussian, but the substantial gap between two distributions leads to degraded sampling quality. FARMER unifies NF and AR within a single framework, using NF component to transform images into latent sequences, whose distribution is implicitly modeled by AR component for easier modeling and controllable sampling. Furthermore, FARMER adopts a self-supervised dimension reduction to partition NF latent channels into distinct groups, making AR modeling feasible and scalable.

In contrast, Autoregressive (AR) models directly factorize sequence likelihoods via the chain rule and lead to the scaling successes of Large Language Models [1, 2, 18, 65, 66]. However, modeling the likelihood over continuous, high-dimensional image pixels remains notably challenging compared with the discrete texts. Continuous AR over visual pixels has been explored for years—from convolutional PixelRNN/PixelCNN [71, 72] to Image Transformer [52] and iGPT [6]. Despite these efforts, continuous AR suffers from extremely long sequences, making training and sampling costly and brittle to long-range dependencies. This gap motivates revisiting how we parameterize continuous densities over high-dimensional pixel spaces and how we couple them with scalable sequence models.

At the same time, Normalizing Flow (NF) [20, 36, 86] has seen a resurgence for image generation. By providing exact likelihoods via invertible and differentiable mappings, NF offers an attractive route for revitalizing continuous AR modeling and a principled latent representation. For instance, JetFormer [70] and STARFlow [20] each design a new NF Transformer as the visual tower: JetFormer employs Jet [36] to enable end-to-end continuous AR modeling over raw image pixels, while STARFlow extends TARFlow [86] and demonstrates that continuous Autoregressive Flow can achieve competitive generation quality. But recent NF works [14, 15, 20, 32, 36, 60, 86] predominantly map the data distribution to a standard Gaussian. This is a challenging objective, as forcing a high-dimensional and highly dispersed data distribution onto a simple isotropic Gaussian can introduce discontinuities or distortions, thus often causing latents sampled from the latent space to become out-of-distribution upon transforming back to the data space, leading to quality degradation.

Inspired by the great work of Jetformer [70], we propose a framework named FARMER that leverages the strengths of both Normalizing Flows and Autoregressive models. As shown in Fig. 1, rather than mapping the data distribution to a fixed standard Gaussian, we employ an NF to transform images into a latent sequence whose distribution is modeled implicitly by an AR model. Concretely, we implement the NF with an Autoregressive Flow (AF) architecture, ensuring causal modeling for NF/AR within FARMER. The two components are optimized jointly in an end-to-end fashion, preserving the tractable, exact likelihoods of NFs while endowing the target distribution with the expressivity of AR modeling. Beyond this design, two inherent challenges remain: (i) **Continuous AR over pixels**: Natural images are highly redundant. Without compression via VAEs [33, 61] or discrete tokenizers [57, 73], directly modeling all pixels forces the AR model to handle extremely long-range pixel dependencies, and thus results in unstable training and sample quality degrading. (ii) **Slow reverse inference in AF**: While AF substantially enhances the mapping capability via next-token modeling, they incur slow inference because the reverse inference process is strictly sequential.

To mitigate the redundancy in pixel AR modeling, we introduce a self-supervised dimension reduction mechanism that partitions NF latent channels into informative and redundant groups without information loss. The key insight is to factorize the token likelihood $P(Z \mid c)$ as

$$\begin{aligned} P(Z \mid c) &= P(Z^R \mid Z^I, c)\, P(Z^I \mid c) \\ &= \Big[\prod_{i=1}^{N} P_{N+1}(Z_i^R \mid Z^I, c)\Big] \Big[\prod_{i=1}^{N} P_i(Z_i^I \mid Z_{<i}^I, c)\Big], \end{aligned}$$

where Z^I denotes the informative channels and Z^R the redundant channels of each token. Concretely, the informative channels Z_i^I are modeled in the standard autoregressive manner, *i.e.*, conditioned on the preceding informative tokens $Z_{<i}^u$ and context c. The redundant channels Z_i^R across all tokens are modeled jointly by a shared distribution conditioned on the entire sequence of informative channels Z^I and context c. This construction allows us to treat the redundant channels of all tokens as a single additional token, effectively converting N high-dimensional tokens into $N{+}1$ lower-dimensional tokens. Maximizing the resulting token likelihood encourages FARMER to disentangle information across channel groups, *i.e.*, concentrating contour and structural features in Z^I, while assigning detail and color information to Z^R, as illustrated in Fig. 6.

For the slow reverse process issue of AF, we propose a one-step distillation scheme for efficient inference, which distills a single-step student reverse path from the teacher's forward path, thereby avoiding the causal reverse process of AF models. Finally, we present a resampling-based Classifier-Free Guidance (CFG) algorithm that significantly improves generation quality in this framework. We summarize our contributions as follows:

- We introduce FARMER, an elegant and powerful framework that jointly optimizes Autoregressive Flow and Autoregressive Transformer for image generation and continuous image pixel likelihood estimation.
- We propose a self-supervised dimension reduction approach simplifying high-dimensional visual modeling.
- We develop a one-step distillation method that accelerates AF reverse process by a factor of $22\times$ with fewer additional training epochs, while maintaining comparable generation quality.
- We introduce a novel resampling-based CFG algorithm that substantially enhances generation quality.

2. Preliminary

This section focuses on the preliminaries of Normalizing Flows and Autoregressive models. A detailed discussion of related works is given in Supplement Sec. 6.

2.1. Normalizing Flows

Normalizing Flows [14, 15, 32, 34, 36, 50, 54, 60, 72, 86] map a complex data distribution $x \sim p_{data}(x)$ into a simple one $z \sim p_Z(z)$. The target distribution $p_Z(z)$ is usually chosen as a standard Gaussian, which is easy for density estimation and sampling. This transformation is achieved by applying a sequence of invertible functions $F = f_n \circ f_{n-1} \circ \cdots \circ f_1$. Accordingly, the forward and inverse mappings are $z = F(x) = f_n \circ f_{n-1} \circ \cdots \circ f_1(x)$ and $x = F^{-1}(z) = f_1^{-1} \circ f_2^{-1} \circ \cdots \circ f_n^{-1}(z)$, respectively.

Using the change-of-variables formula, NFs can calculate the exact probability density of a data point x as:

$$p_{data}(x) = p_Z(z) \left|\det\left(\frac{\partial z}{\partial x}\right)\right| = p_Z(F(x)) \left|\det\left(\frac{\partial F(x)}{\partial x}\right)\right|, \quad (1)$$

where $\det\left(\frac{\partial F(x)}{\partial x}\right)$ is the Jacobian determinant of F. NFs are trained via maximum likelihood estimation, formulated in terms of Negative Log-Likelihood (NLL):

$$\min_{F} -\log p_Z(F(x)) - \log\left|\det\left(\frac{\partial F(x)}{\partial x}\right)\right|. \quad (2)$$

Previous works [20, 86] consider p_Z as a standard Gaussian distribution $\mathcal{N}(0,1)$, so Eq. (2) can be written as $\min_F\ 0.5\,||F(x)|\,|_2^2 - \log\left|\det\left(\frac{\partial F(x)}{\partial x}\right)\right|$.

2.2. AutoRegressive Models

AutoRegressive models formulate the likelihood $p(z)$ of a token sequence $z = (z_1, z_2, \ldots, z_N)$ by factorizing it into a product of next-token conditional probabilities: $\prod_{i=1}^{N} p(z_i|z_{<i})$, where AR conditions only on the previous tokens $z_{<i} = (z_1, \ldots, z_{i-1})$ to predict the next token z_i. Such AR paradigm has achieved remarkable scalability and tremendous success in language models [1, 2, 18, 65, 66]. Furthermore, it has also demonstrated promising capabilities in visual generation [23, 40, 44, 64, 68].

3. Approach

3.1. Mapping Images to AR Distributions via NFs

As mentioned in Sec. 2.1, mapping the high-dimensional and highly dispersed image data distribution to a simple isotropic Gaussian distribution via an NF can induce out-of-distribution issues and degrade the sampling quality [20]. Inspired by JetFormer [70], we propose a framework that combines the strengths of NF and AR models. Rather than using a fixed standard Gaussian, we employ an NF to transform images into a latent sequence whose distribution is modeled implicitly by an AR model. Then the NF and AR components are optimized jointly in an end-to-end fashion, preserving the tractable, exact likelihoods of NFs while endowing the target distribution with the expressivity of AR modeling. The objective Eq. (2) is formulated as:

$$\min_{F,AR} -\sum_{i=1}^{N} \log p_{AR}(z_i|z_{<i}) - \log\left|\det(\frac{\partial F(x)}{\partial x})\right|, \quad (3)$$

where $z = F(x)$ denotes the forward transformation of the NF. The target distribution over z is parameterized autoregressively. To enhance the expressivity of the AR base, following JetFormer and GIVT [69], we model each conditional probability $p(z_i|z_{<i})$ with a K-component Gaussian Mixture Model. Furthermore, different from JetFormer, we implement the NF component $F(x)$ as an Autoregressive Flow (AF) [34, 50, 86]. This design ensures the entire pipeline maintains a consistent and powerful causal formulation. Notably, when $K = 1$, our model, composed of an AF and an AR model, reduces to a single, deeper AF (see Supplement Sec. 9.1).

3.2. Flow AutoRegressive Transformer

We devise Flow AutoRegressive transforMER models (FARMER) that unify an invertible autoregressive flow with an autoregressive model into a single framework, which enables end-to-end training on raw image pixels by mapping the data onto an implicit distribution modeled by the AR.

Dequantize and Patchify. Given an input image $I \in \mathbb{R}^{H\times W\times C}$, following previous works [53, 70, 86], FARMER first add a small Gaussian noise $\mathcal{N}(0,\sigma^2)$ to the raw image I to dequantize the discrete pixel values and create a more continuous data distribution. Following JetFormer [70], we enhance this technique by employing an annealed noise strategy, where the noise level σ is annealed from 0.1 to 0.005 using a cosine decay schedule. Then we patchify the noised image with a downsampling factor p into a representation $I' \in \mathbb{R}^{h\times w\times d}$, where $h = H/p$, $w = W/p$, and $d = C\cdot p^2$. Finally, we reshape I' into a sequence of $N = h\cdot w$ continuous-valued visual tokens $x = \{x_1, x_2, \ldots, x_N\}$, where each token $x_i \in \mathbb{R}^d$. Notably, there is no dimension compression in this process.

Forward and Reverse of Autoregressive Flow. During training, FARMER utilizes an autoregressive flow F to map token sequence $x \in \mathbb{R}^{N\times d}$ to latents $z \in \mathbb{R}^{N\times d}$, *i.e.*, $z = F(x)$. By design, F is invertible (see Fig. 2) and composed of n invertible blocks: $F = f_n \circ f_{n-1} \circ \cdots \circ f_1$. Letting $z^0 = x$ and $z^n = z$, the forward transformation for the t-th AF block, $z^t = f_t(z^{t-1})$, is defined for each token z_i^t [1] as:

$$z_i^t = \begin{cases} z_1^{t-1} & \text{if } i = 1, \\ (z_i^{t-1} - \mu_t(z_{<i}^{t-1})) \odot \sigma_t(z_{<i}^{t-1}) & \text{if } i > 1, \end{cases} \quad (4)$$

where $z_{<i}^{t-1}$ denotes the preceding tokens $\{z_1^{t-1}, ..., z_{i-1}^{t-1}\}$. The bias factor $\mu_t(z_{<i}^{t-1})$ and the scaling factor $\sigma_t(z_{<i}^{t-1})$ are predicted by the t-th block conditioned on $z_{<i}^{t-1}$ in a causal manner. Accordingly, the inverse transformation of the t-th block, $z^{t-1} = f_t^{-1}(z^t)$, is derived as (as shown in Fig. 3a):

$$z_i^{t-1} = \begin{cases} z_1^t & \text{if } i = 1, \\ (z_i^t \oslash \sigma_t(z_{<i}^{t-1})) + \mu_t(z_{<i}^{t-1}) & \text{if } i > 1, \end{cases} \quad (5)$$

where $\oslash$ denotes element-wise division. Notably, for each block t, the causal formula of forward transformations ensures that the Jacobian $\frac{\partial z^t}{\partial z^{t-1}}$ is lower triangular. Thus, its determinant equals the product of its diagonal entries, σ_t and can be computed efficiently during training. By the chain rule, the total log-det of F is the sum over blocks:

$$\log\left|\det\frac{\partial z}{\partial x}\right| = \sum_{t=1}^{n}\log\left|\det\frac{\partial z^t}{\partial z^{t-1}}\right| = \sum_{t=1}^{n}\sum_{i=1}^{N}\sum_{j=1}^{d}\log\left|[\sigma_t(z_{<i}^{t-1})]_j\right|. \quad (6)$$

Permutation. To improve AF expressiveness [86], we apply a permutation π_t (which reverses the token order) to

[1] Subscripts denote indexing i along the token sequence dimension, and superscripts denote the t-th AF block indices.

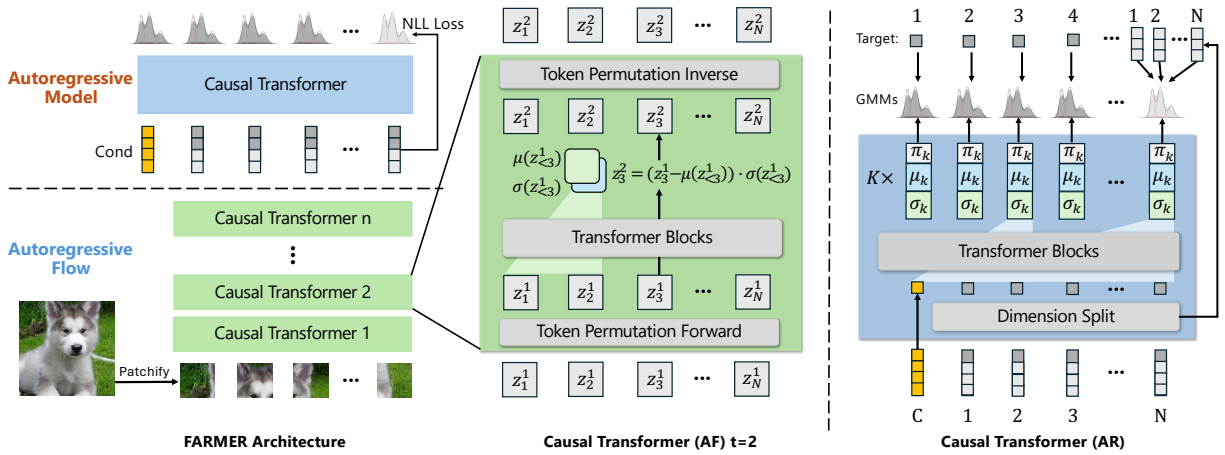


Figure 2. **Overview of FARMER. Left**, FARMER consists an autoregressive flow (AF) and an autoregressive (AR) model. The AF maps image patches to latent sequences, while the AR predicts Gaussian Mixture Models (GMMs) conditioned on these latents, optimizing their likelihood end-to-end. **Middle**, Each AF block performs an invertible next-token transformation of the input sequence to obtain a new sequence. **Right**, AR splits latent channels into informative and redundant groups, modeling each informative token's likelihood via a GMM conditioned on its previous tokens, and redundant tokens jointly via a shared GMM conditioned on all informative tokens. This separation enables disentangling structural and detailed information.

z^{t-1} at the start of each block t. After the forward AF transformation $z^t = f_t(z^{t-1})$, we apply the inverse permutation π_t^{-1} to z^t to restore the original ordering (see Fig. 2).

AR Modeling. After the AF forward mapping, we get the latent representation $z = \{z_1, z_2, ..., z_N\}$ from the input image. Then we model its probability distribution with a large causal AR Transformer. The AR Transformer is conditioned on an embedding $c \in \mathbb{R}^{1\times D}$ which encodes conditional information such as a class label. To amplify its effect, we replicate the condition embedding M times and prepend it to the latent sequence z. By the chain rule, $P(z|c) = \prod_{i=1}^{N} p(z_i|z_{<i}, c)$. For each token, the AR Transformer predicts the parameters of a K-component Gaussian Mixture Model (GMM) distribution G_i:

$$p(z_i|z_{<i}, c) = \sum_{k=1}^{K} \pi_k(z_{<i}, c)\,\mathcal{N}\big(z_i\,;\,\mu_k(z_{<i}, c), \mathrm{diag}(\sigma_k^2(z_{<i}, c))\big), \quad (7)$$

where $\pi_k \in \mathbb{R}, \mu_k, \sigma_k \in \mathbb{R}^d$ are mixture weights, means, and standard deviations of the k-th GMM component.

Learning Objective. Combining the AR likelihood (Eq. (7)) and the AF log-determinant (Eq. (6)) to Eq. (3), the training loss of FARMER is the negative log-likelihood (NLL) of data and average over all dimensions:

$$\mathcal{L} = -\frac{1}{N\cdot d}\left(\sum_{i=1}^{N}\log p(z_i|z_{<i}, c) + \log\left|\det\frac{\partial z}{\partial x}\right|\right). \quad (8)$$

3.3. Self-supervised Dimension Reduction

A fundamental challenge in pixel AR modeling is redundancy: natural images are intrinsically low-dimensional signals whose spectrum is dominated by low frequencies [70]. Although an invertible AF can faithfully map the data distribution, its bijective nature preserves dimensionality. For a $256 \times 256 \times 3$ image with patch size 16, the latent sequence has $N = (\frac{256}{16})^2 = 256$ tokens, each with dimension $d = 768$. This high-dimensional latent Z exacerbates two issues: (i) per-token AR modeling with a K-component GMM in $\mathbb{R}^d$ becomes exceptionally challenging. (ii) The enlarged latent volume expands the sampling space, reducing sampling efficiency and often degrading sample quality.

Prior work like RealNVP [15] factors out half of the dimensions and models them with Gaussian priors. JetFormer [70] adopts a similar strategy: it models the informative dimensions Z^I autoregressively and assigns the redundant dimensions to Z^R a standard Gaussian prior, assuming $P(Z \mid c) = P(Z^R)\,P(Z^I \mid c)$, i.e., Z^R is independent of both Z^I and c. This is a strong assumption that is often violated in practice: informative and redundant parts typically remain correlated, so enforcing independence can discard information. Moreover, decoupling Z^R from c and Z^I restricts how other modalities interact with the full latents, leading to potential suboptimal performance on multimodal tasks.

To this end, we propose a novel self-supervised dimension reduction technique to address the above issues. It reduces the complexity of AR modeling, reduces the sampling space, and reduces computational cost, all while avoiding information loss. As shown in Fig. 2, we split the latent $Z \in \mathbb{R}^{N\times d}$ channel-wise into an informative part $Z^I \in \mathbb{R}^{N\times d^I}$ and a redundant part $Z^R \in \mathbb{R}^{N\times d^R}$, with $d = d^I + d^R$. Then we correctly factorize the joint probability via the chain rule:

$$P(Z \mid c) = P(Z^I \mid c)\,P(Z^R \mid Z^I, c).$$

Rather than assuming that Z^R is independent of (Z^I, c) in JetFormer, we explicitly condition Z^R on both c and Z^I,

where Z^I serves as the global image context. Furthermore, we constrain all tokens in Z^R to share a GMM distribution, while modeling tokens in Z^I in a token-by-token manner. This design encourages self-supervised disentanglement of distinct information across channel groups, without relying on a standard Gaussian prior.

For $P(Z^I|c)$, we model each informative token Z_i^I autoregressively with an individual GMM distribution G_i predicted by the AR Transformer conditioned on c and the preceding $Z_{<i}^I$, thereby being capable of capturing complex distributions. In contrast, for $P(Z^R|Z^I, c)$, we use the entire informative sequence Z^I (global context) together with c to predict a single shared GMM G_{N+1} for all redundant tokens Z_i^R. By maximizing the combined likelihoods, our method successfully encourages complex contour and structural information to be isolated in Z^I, while the simple color and fine-detail information is allocated to Z^R, as shown in Fig. 6 and discussed in Sec. 4.3.

After dimension reduction, the final training loss $\mathcal{L}$ is rewritten as the sum of the NLL for both components:

$$\mathcal{L} = -\frac{1}{N \cdot D}\left(\sum_{i=1}^{N} \log p(z_i^I | z_{<i}^I, c) + \sum_{i=1}^{N} \log p(z_i^R | z_{\leq N}^I, c) + \log \left|\det \frac{\partial z}{\partial x}\right|\right). \tag{9}$$

3.4. Resampling-based Classifier-Free Guidance

Classifier-Free Guidance (CFG) has become a standard technique for improving sample quality in diffusion models [45, 55, 61] and visual autoregressive models [40, 64, 68]. Conceptually, CFG steers the sampling process from a base distribution towards a target conditional distribution. For FARMER, the guided log-probability for a latent token z can be formulated as:

$$\begin{aligned} \log p'(z) &\propto \log p_c(z) + w \cdot (\log p_c(z) - \log p_u(z)) \\ &= \log p_u(z) + (w+1) \cdot (\log p_c(z) - \log p_u(z)), \end{aligned} \tag{10}$$

where $p_c(z) = p(z|c)$ and $p_u(z) = p(z|\emptyset)$ are the conditional and unconditional GMM, and w is the guidance scale. However, the guided distribution $p'(z)$ is an intractable mixture of GMMs, making direct sampling infeasible.

To make it practical, we introduce a novel Resampling-based CFG. The key insight is that the target distribution $p'(z)$ can be decomposed into two components as shown in Eq. (10): the first term (blue) is a tractable GMM distribution and can be sampled directly, while the second term (red) is not samplable but allows evaluation of the sample probability under this distribution. Therefore, we approximate the sampling from $p'(z)$ via a three-step resampling scheme detailed in Supplement Algorithm 1. For each token z_i, the procedure is: (i) **Propose**. Sample s candidates from the conditional GMM $p_c(z_i)$ and s' candidates from the unconditional GMM $p_u(z_i)$ respectively. (ii) **Weigh**. Compute the corresponding log probability of all candidates as the second term in Eq. (10), and normalize the weights. (iii) **Resample**. Resample from the categorical distribution that consists of the normalized weights of all candidates, to obtain the final sample. In summary, the probability of candidate z is selected in the "propose" step is $p_c(z)/p_u(z)$, and that in the "resample" step is $\left(\frac{p_c(z)}{p_u(z)}\right)^w / \left(\frac{p_c(z)}{p_u(z)}\right)^{w+1}$. This resampling procedure ensures that the overall probability $p_c(z)\left(\frac{p_c(z)}{p_u(z)}\right)^w$ matches the target probability $p'(z)$. More details are provided in the Supplement Sec. 8.2.

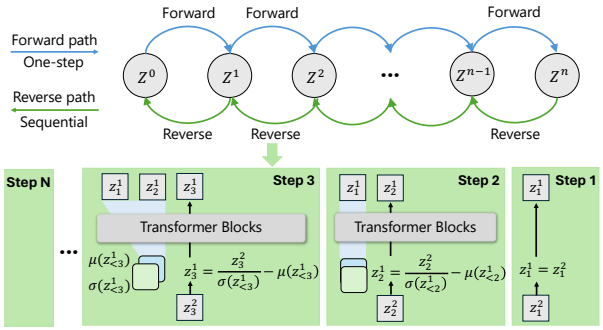
3.5. Fast Inferring via One-Step Distillation

A significant drawback of Autoregressive Flows is the slow inference speed, due to its sequential reverse process. As shown in Eq. (5), during the inverse mapping f_t^{-1} of AF block t, the calculation of each token $z_{t-1,i}$ is conditioned on preceding tokens $z_{t-1,<i}$. Such dependency brings a substantial inference speed bottleneck, which is also noted in recent AF works like TARFlow [86] and STARFlow [20] whose token sequence length is 1024 with a patchsize of 8.

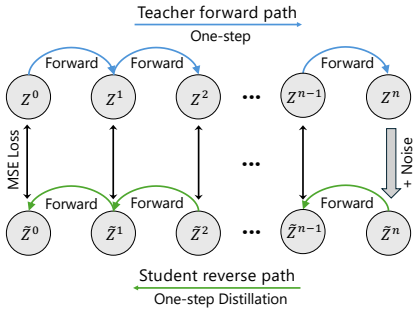
Benefiting from Normalizing Flow invertibility, whose forward and reverse paths are exact inverses, we can train a new AF whose forward path mirrors the original AF's reverse path. Furthermore, because the forward/reverse path of NF consists of finite steps, we can invert the original AF's forward path $(Z_0, Z_1, ..., Z_n)$ to obtain its reverse path $(Z_n, Z_{n-1}, ..., Z_0)$, and utilize such reverse path to supervise the new AF, thereby avoiding the original AF to perform slow reverse process to obtain its reverse path.

As shown in Fig. 3 and inspired by the generative distillation works [62, 74, 81], we propose a one-step distillation scheme that learns a single-step student reverse path from the trained teacher's forward path. Supplement Algorithm 2 details the procedure: we first obtain a teacher AF model, trained within the FARMER framework. Then, we initialize the student by copying the teacher AF and enable its attention bidirectional. At each distill iteration, we input training data z_0 to the teacher AF to obtain a teacher forward path $F(Z^0) = (Z^0, Z^1, ..., Z^n)$ and use its reversal $(Z^n, Z^{n-1}, ..., Z^0)$ as the supervision target for the student's forward path $G(\tilde{Z}^n) = (\tilde{Z}^n, \tilde{Z}^{n-1}, ..., \tilde{Z}^0)$. For robustness, we use a noised latent ($\tilde{Z}^n = Z^n + s \cdot noise$) as the student AF's input. Then, the output $\tilde{Z}^{t-1}$ of each student AF block t is supervised by minimizing the Mean Squared Error (MSE) against the Z^{t-1} from the teacher path. By distilling one such student AF model, we significantly accelerate the reverse process from 0.1689 to 0.0076 seconds per image. As discussed in Sec. 4.3 and Tab. 4, such one-step distillation brings a 22× acceleration for AF reverse process while maintaining comparable generation quality.

Notably, unlike progressive diffusion distillation, our approach distills the entire AF model in an end-to-end manner, ensuring robustness to cumulative inference error; it eliminates the need for teacher models to run the inference



(a) Autoregressive Flow Reverse Process



(b) One-Step Distillation Process

Figure 3. **One-Step Distillation.** (a) The autoregressive flow (AF) reverse process reconstructs tokens sequentially, conditioning each token on previous ones, which leads to slow inference. (b) Our method distills a one-step student reverse path from the frozen teacher forward path in an end-to-end manner, approximating the reverse process of each AF block by the corresponding student AF block's forward process, thereby enabling 22× faster AF reverse process and 4× overall inference speed-up.

process, thereby accelerating the distillation process; and it requires fewer additional training epochs on the AF.

4. Experiments

4.1. Experimental Settings

We verify our proposed FARMER on ImageNet [12] at 256 × 256 resolution. We evaluate Fréchet Inception Distance (FID) [24], Inception Score (IS) [63] and Precision/Recall [37] on 50K generated samples. We design two model scales: FARMER-1.1B/1.9B. By default, for the GMM prediction heads, the informative dimensions (d^I) are set to 128 with $K = 64$ mixtures, while redundant dimensions (d^R) are set to 640 with $K = 200$. The network architecture and training setup are detailed in Supplement Sec. 8.1.

4.2. Results

System-level Comparison. As shown in Tab. 1, we compare FARMER with various generative models, including both latent and pixel-based approaches. Notably, FARMER significantly outperforms JetFormer [70], the most comparable baseline to our model, reducing the FID by 3.04. Furthermore, FARMER demonstrates superior generation quality compared to the NF-based models, TARFlow [86] and STARFlow [20]. FARMER also achieves competitive performance and faster convergence speed against mainstream Generative Adversarial Networks (GANs), diffusion models, and AR models. While methods like PixelFlow [7] and PixNerd [77] employ complex multi-stage pipelines to achieve better results, our approach remains highly competitive by utilizing a simple, single-stage, end-to-end training strategy. Compared to latent generative models, our method maintains strong generative performance. Latent generative models often benefit from a well-structured continuous latent space, modeled by VAEs, that facilitates high-quality sampling. However, by operating directly in pixel space, our model gains direct access to the raw data distribution,

Figure 4. **Qualitative Comparison**. Images of class 0 in ImageNet generated by FARMER, MAR, and DiT.

which can potentially capture more detailed data semantics without the information bottleneck imposed by VAEs.

Qualitative Results. We show qualitative results generated by FARMER-1.9B using resampling-based CFG in Supplement Sec. 7. FARMER generates diverse images with high quality. A key advantage of FARMER over latent generative models is its ability to preserve fine-grained details. This is because our end-to-end training directly accesses the raw data distribution, and the invertible nature of NFs prevents information loss. As shown in Fig. 4, our FARMER can reconstruct intricate features, such as faces, which are often blurred or distorted by the compression of VAEs.

4.3. Experimental Analysis

Ablation Study. Here we investigate the impact of each component within the FARMER framework on overall performance. As shown in Tab. 2, all experiments are conducted using FARMER-1.1B with K=1024 GMM mixtures on ImageNet 256×256 for class-conditional image generation. Natural images typically possess a high degree of redundancy, and low-dimensional signals with low-frequency components dominating the spectrum [70]. Direct transformation of original images using normalizing flows yields latent representations with unchanged dimensionality. Partitioning these high-dimensional latents into equal-length, high-dimensional tokens complicates AR modeling and sampling. By introducing a self-supervised dimension re-

Table 1. **System performance comparison** on ImageNet 256 × 256 class-conditioned generation. "↓"/"↑" indicate lower/higher values are better. Metrics include FID, IS, precision and recall. **Resampling-based CFG** is applied on FARMER.

Types	Models	Params	Epochs	FID↓	IS↑	Pre.↑	Rec.↑
Latent Generative Models							
Diff.	LDM-4 [61]	400M + 86M	170	3.6	247.7	0.87	0.48
	DiT-XL [55]	675M + 86M	1400	2.27	278.2	0.83	0.57
	SiT-XL [45]	675M + 86M	1400	2.06	270.3	0.82	0.59
	FlowDCN [76]	618M + 86M	400	2.00	263.1	0.82	0.58
	REPA [85]	675M + 86M	800	1.42	305.7	0.80	0.64
	DDT-XL [78]	675M + 86M	400	1.26	310.6	0.79	0.65
	REPA-E [39]	675M + 86M	800	1.12	302.9	0.79	0.66
AR	GIVT [69]	1.67B+53M	500	2.59	-	0.81	0.57
	MAR-AR [40]	479M+66M	800	4.69	244.6	-	-
	MAR-L [40]	479M + 66M	800	1.78	296.0	0.81	0.60
NF	STARFlow [20] one-step denoise	1.4B+86M	320	2.96	-	-	-
	STARFlow [20] finetune decoder	1.4B+86M	320	2.40	-	-	-
Pixel Generative Models							
GAN	BigGAN [4]	112M	-	6.95	224.5	0.89	0.38
Diff.	ADM [13]	554M	400	4.59	186.7	0.82	0.52
	CDM [26]	-	2160	4.88	158.7	-	-
	SimpleDiffusion [27]	2.0B	800	2.77	211.8	-	-
	PixelFlow-XL/4 [7]	677M	320	1.98	282.1	0.81	0.60
	PixNerd-XL/16 [77]	700M	320	1.93	298	0.80	0.60
	SiD2 patch 1 [28]	-	1280	1.38	-	-	-
AR	FractalMAR-H [41]	844M	600	6.15	348.9	0.81	0.46
NF	TARFlow [86] patch 8	1.3B	320	5.56	-	-	-
	STARFlow [20] patch 8	1.4B	320	4.69	-	-	-
NF+AR	JetFormer [70]	2.8B	500	6.64	-	0.69	0.56
	FARMER 1.1B patch 16	1.1B	320	5.40	212.23	0.78	0.45
	FARMER 1.1B patch 8	1.1B	320	5.02	237.00	0.80	0.45
	FARMER 1.9B patch 16	1.9B	320	3.96	250.64	0.79	0.50
	FARMER 1.9B patch 8	1.9B	320	3.60	269.21	0.81	0.51

Table 2. **Ablation study of FARMER.** We demonstrate relative impact of various components on generation quality.

Self. Dim. Reduce	Cond. Repeat	Final Permute	CFG Method	FID↓	IS↑
✗	✗	✗	✗	61.17	22.10
✓	✗	✗	✗	49.29	30.61
✓	✓	✗	✗	45.34	33.87
✓	✗	✓	✗	45.69	33.73
✓	✓	✓	✗	44.56	33.17
✓	✓	✓	Naive Method	8.66	233.84
✓	✓	✓	Resampling-based	5.67	215.53

Table 3. **Impact of Normalizing Flow Architectures**.

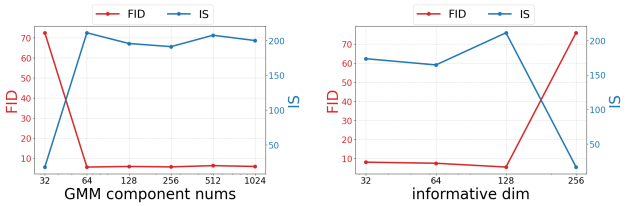
NF Architectures	FID↓	IS↑	Forward Speed	Reverse Speed
Jet	106.23	13.14	0.0065 s/img	0.0099 s/img
AF	5.55	194.63	0.0066 s/img	0.1689 s/img
AF+One-step Distll.	5.63	193.49	0.0066 s/img	0.0076 s/img

duction design as Eq. (9), the FID notably decreases from 61.17 to 49.29, and IS also improves from 22.10 to 30.61. Next, we repeat the class embedding 64 times to enhance the conditional guidance, the FID further decreases to 45.34. If we consider the AR model as a block of AF, adding a token permutation operation between AF and AR is beneficial to preserve the fixed dependency between token sequences. The FID further decreases to 44.56. CFG is essential for improving generation quality in modern generative models during sampling. We first adopt a naive CFG sampling method from JetFormer [70], the FID score notably decreases to 8.66. Then, we upgrade the CFG sampling method to a resampling-based method described in Sec. 3.4, and the FID score further decreases to 5.67. Together, these designs enable FARMER to achieve strong performance.

Normalizing Flow Architecture Comparison. The architecture of NFs critically affects representational capacity, training, and inference efficiency [14, 15, 32, 34, 36, 50, 60, 72, 86]. Here we primarily compare two architectures, Jet [36] and AF [34, 50], which have demonstrated strong performance in modern generative models Jetformer [70] and TARFlow [86], respectively. For fairness, both models utilize same block numbers, layers per block, and AR modules with similar parameters. Their representational capacity is assessed by FID/IS, while forward and reverse speeds are also reported. Tab. 3 summarizes these results. Specifically, in both forward and reverse processes, Jet applies an affine transform from half of the latent channels to the other half in each block, stacking N such blocks for the full model. This simple and efficient design enables Jet to achieve fast forward and reverse computations, but it also limits its representational capacity, leading to a failure to separate different information of the image into two channel groups. In contrast, AF updates each token sequentially based on previous tokens, enhancing expressiveness but incurring slow reverse process as described in Sec. 3.2. To overcome this, we introduce one-step distillation: a student AF model is distilled from a frozen teacher AF model, requiring only 60 additional epochs. This significantly accel-

erates reverse process from 0.1689 seconds to 0.0076 seconds per image, providing a fast and expressive NF architecture for both training and inference.

Dimension Reduction Method Comparison. We also compare our self-supervised dimension reduction method with the approach adopted in JetFormer [70]. Our method achieves improved generative performance, reducing FID from 7.81 to 5.67, and increasing IS from 182.87 to 215.53.



(a) Impact of GMM mixture num (b) Impact of informative dim

Figure 5. **The ablation study of different properties.**

Impact of GMM Mixture Component Number. We analyze the impact of the number of GMM mixtures predicted by AR models, which reflect the complexity of the approximated distribution. A larger number of mixtures enables the model to represent more complex distributions; however, it also increases sampling difficulty and training computational costs. As shown in Fig. 5a, the FID varies only slightly across different mixture numbers and attains its optimal value at 64 mixtures. Notably, reducing the number further—to 32 mixtures—prevents the model from performing effective dimension reduction, resulting in a significant decline in generation quality. Thus, 64 mixtures are chosen to best balance quality and efficiency.

Impact of the Informative Dimension. We analyze the impact of the informative dimension, which reflects how information is separated and allocated by the NF models. As shown in Fig. 5b, the FID initially decreases as the informative dimension increases and achieves the optimal value at 128. Further increasing the dimension leads to a rise in FID. This phenomenon demonstrates a trade-off: increasing the informative dimension allows capturing more information, but also makes AR modeling and sampling more challenging. Therefore, we set the informative dimension to 128.

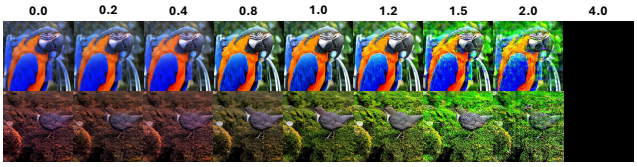


Figure 6. **The impact of redundant channels**. The numbers above indicate scaling factors applied to the variance of the shared GMM distribution for redundant channels.

Information Separation of Two Dimension Groups. Here, we visualize the information contained in the informative and redundant channels. During inference, tokens for informative channels are sequentially predicted, followed by sampling redundant channels from a shared GMM conditioned on the informative tokens. By varying the GMM component variances, we control the diversity of redundant channel samples. As shown in Fig. 6, lower variance leads to sampled tokens clustering near Gaussian means, producing smoother color regions and reduced diversity, while the global structure of the images remains largely unaffected. Higher variance increases diversity but raises the risk of out-of-distribution samples, which can lead to color artifacts or, at extremes, incoherent images. These observations demonstrate that our self-supervised dimension reduction method successfully decouples structural contour information from fine color details.

Table 4. **Inference Speed Accelerate**.

Method	Epochs	FID↓	IS↑	AR infer. time (% in total)	NF reverse time (% in total)	Total time
FARMER	280	5.55	194.63	0.0500s(22.8%)	0.1689s(77.2%)	0.2189s
w/. Distll.	280+60	5.63	193.49	0.0500s(88.2%)	0.0076s(13.4%)	0.0567s

Inference Speed Acceleration. As reported in Table 4, the baseline FARMER model requires 0.2189 seconds per image for inference, with the NF reverse process constituting the majority of this time (77.2%). Applying one-step distillation dramatically reduces the NF reverse time from 0.1689 to 0.0076 seconds, yielding a 22× acceleration for this component. The total inference time decreases from 0.2189 to 0.0567 seconds per image, an almost 4× acceleration, while maintaining comparable image quality (FID 5.63 vs. 5.55, IS 193.49 vs. 194.63). This demonstrates that one-step distillation effectively eliminates the sequential bottleneck of NF reverse process, enabling FARMER to achieve both high fidelity and efficient generation.

5. Conclusion

We introduce FARMER, a novel generative framework that integrate invertible AF with AR model, enabling end-to-end training directly on raw image pixels. FARMER learns by mapping the data distribution to an distribution modeled by the AR model and maximizing the negative log-likelihood of the raw images. This design permits both high-quality image synthesis and explicit likelihood estimation. Furthermore, we propose key techniques: a self-supervised dimension reduction to alleviate the complexity of AR modeling/sampling, a resampling-based CFG strategy to enhance image quality, and a one-step distillation scheme to accelerate the inference speed. Through the contributions, FARMER demonstrates competitive performance in image generation relative to pixel-based and latent generative models. However, beyond the curse of high-dimensionality that we have addressed, two challenges persist in NF–AR, *i.e.*, (i) dequantization relying on noise injection and (ii) the complications arising from the log-determinant loss. We leave these for future works.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 3

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1, 3

[3] Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *Advances in Neural Information Processing Systems*, 37:73952–73985, 2024. 2

[4] Andrew Brock. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 7

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 1

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*. PMLR, 2020. 1

[7] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025. 6, 7

[8] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. 1

[9] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 2

[10] 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. 1

[11] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024. 1

[12] 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*, pages 248–255. Ieee, 2009. 6

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 7

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2, 7, 1

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017. 2, 4, 7

[16] Felix Draxler, Peter Sorrenson, Lea Zimmermann, Armand Rousselot, and Ullrich Köthe. Free-form flows: Make any architecture a normalizing flow. In *International Conference on Artificial Intelligence and Statistics*, pages 2197–2205. PMLR, 2024.

[17] Felix Draxler, Stefan Wahl, Christoph Schnörr, and Ullrich Köthe. On the universality of volume-preserving and coupling-based normalizing flows. *arXiv preprint arXiv:2402.06578*, 2024. 1

[18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1, 3

[19] Robert Giaquinto and Arindam Banerjee. Gradient boosted normalizing flows. *Advances in Neural Information Processing Systems*, 33:22104–22117, 2020. 1

[20] Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *arXiv preprint arXiv:2506.06276*, 2025. 2, 3, 5, 6, 7

[21] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024. 1

[22] Tiankai Hang, Jianmin Bao, Fangyun Wei, and Dong Chen. Fast autoregressive models for continuous latent generation. *arXiv preprint arXiv:2504.18391*, 2025. 1

[23] Wanggui He, Siming Fu, Mushui Liu, Xierui Wang, Wenyi Xiao, Fangxun Shu, Yi Wang, Lei Zhang, Zhelun Yu, Haoyuan Li, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17123–17131, 2025. 3

[24] 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. 6

[25] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International conference on machine learning*, pages 2722–2730. PMLR, 2019. 2

[26] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 7

[27] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 7

[28] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024. 7

[29] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International conference on machine learning*, pages 2078–2087. PMLR, 2018. 1

[30] Guolin Ke and Hui Xue. Hyperspherical latents improve continuous-token autoregressive generation. *arXiv preprint arXiv:2509.24335*, 2025. 1

[31] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 2

[32] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 2, 7, 1

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[34] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016. 2, 3, 7, 1

[35] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020. 1

[36] Alexander Kolesnikov, André Susano Pinto, and Michael Tschannen. Jet: A modern transformer-based normalizing flow. *arXiv preprint arXiv:2412.15129*, 2024. 2, 7, 1

[37] 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. 6

[38] 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*, pages 11523–11532, 2022. 1

[39] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025. 7

[40] 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. 3, 5, 7, 1

[41] Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *arXiv preprint arXiv:2502.17437*, 2025. 7, 2

[42] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025. 1

[43] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*. 2

[44] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024. 3

[45] 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. *arXiv preprint arXiv:2401.08740*, 2024. 5, 7

[46] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751, 2025. 1

[47] Bálint Máté, Samuel Klein, Tobias Golling, and François Fleuret. Flowification: Everything is a normalizing flow. *Advances in Neural Information Processing Systems*, 35: 35478–35489, 2022. 1

[48] Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in neural information processing systems*, 33:2503–2515, 2020. 1

[49] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 2

[50] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. 2, 3, 7, 1

[51] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 1

[52] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*. PMLR, 2018. 1

[53] Marco Pasini, Javier Nistal, Stefan Lattner, and George Fazekas. Continuous autoregressive models with noise augmentation avoid error accumulation. *arXiv preprint arXiv:2411.18447*, 2024. 3, 1

[54] Massimiliano Patacchiola, Aliaksandra Shysheya, Katja Hofmann, and Richard E Turner. Transformer neural autoregressive flows. *arXiv preprint arXiv:2401.01855*, 2024. 2, 1

[55] 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. 5, 7

[56] 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*. Pmlr, 2021. 1

[57] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2, 1

[58] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. *arXiv preprint arXiv:2412.15205*, 2024. 1

[59] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025. 1

[60] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 2, 7, 1

[61] 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*, pages 10684–10695, 2022. 2, 5, 7

[62] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 5

[63] 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. 6

[64] 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. 3, 5, 1, 2

[65] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 3

[66] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 1, 3

[67] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou, Haomiao Tang, et al. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025. 1

[68] 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. 3, 5, 1

[69] Michael Tschannen, Cian Eastwood, and Fabian Mentzer. GIVT: Generative infinite-vocabulary Transformers. *arXiv:2312.02116*, 2023. 3, 7, 1

[70] Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. Jetformer: An autoregressive generative model of raw images and text. *arXiv preprint arXiv:2411.19722*, 2024. 2, 3, 4, 6, 7, 8, 1

[71] 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. 1, 2

[72] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*. PMLR, 2016. 1, 2, 7

[73] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2, 1

[74] Steven Walton, Valeriy Klyukin, Maksim Artemev, Denis Derkach, Nikita Orlov, and Humphrey Shi. Distilling normalizing flows. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 5, 1

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

[76] Shuai Wang, Zexian Li, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Exploring dcn-like architecture for fast image generation with arbitrary resolution. *Advances in Neural Information Processing Systems*, 37:87959–87977, 2024. 7

[77] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025. 6, 7

[78] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025. 7

[79] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. 1

[80] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025. 1

[81] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024. 5

[82] Hu Yu, Hao Luo, Hangjie Yuan, Yu Rong, and Feng Zhao. Frequency autoregressive image generation with continuous tokens. *arXiv preprint arXiv:2503.05305*, 2025. 1

[83] 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*, 2022. 1

[84] Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36: 78808–78823, 2023. 2

[85] 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. 7

[86] Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024. 2, 3, 5, 6, 7, 1

[87] Ruixiang Zhang, Shuangfei Zhai, Jiatao Gu, Yizhe Zhang, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Josh Susskind, and Navdeep Jaitly. Flexible language modeling in continuous space with transformer-based autoregressive flows. *arXiv preprint arXiv:2507.00425*, 2025. 1

[88] Qinyu Zhao, Stephen Gould, and Liang Zheng. Arinar: Bi-level autoregressive feature-by-feature generative models. *arXiv preprint arXiv:2503.02883*, 2025. 1

[89] Guangting Zheng, Yehao Li, Yingwei Pan, Jiajun Deng, Ting Yao, Yanyong Zhang, and Tao Mei. Hierarchical masked autoregressive models with low-resolution token pivots. *arXiv preprint arXiv:2505.20288*, 2025. 1