# Autoregressive Distillation of Diffusion Transformers

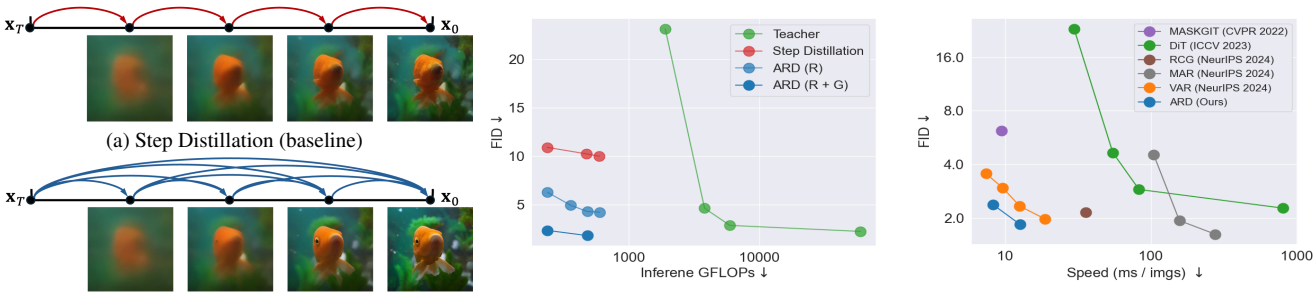
Yeongmin Kim[†,‡,*] Sotiris Anagnostidis[†,§] Yuming Du[†] Edgar Schönfeld[†] Jonas Kohler[†]
Markos Georgopoulos[†] Albert Pumarola[†] Ali Thabet[†] Artsiom Sanakoyeu[†]

Figure 1. Samples ($1024 \times 1024$) generated by our 3-step ARD model, distilled from a 1.7B Emu.

## Abstract

*Diffusion models with transformer architectures have demonstrated promising capabilities in generating high-fidelity images and scalability for high resolution. However, iterative sampling process required for synthesis is very resource-intensive. A line of work has focused on distilling solutions to probability flow ODEs into few-step student models. Nevertheless, existing methods have been limited by their reliance on the most recent denoised samples as input, rendering them susceptible to exposure bias. To address this limitation, we propose AutoRegressive Distillation (ARD), a novel approach that leverages the historical trajectory of the ODE to predict future steps. ARD offers two key benefits: 1) it mitigates exposure bias by utilizing a predicted historical trajectory that is less susceptible to accumulated errors, and 2) it leverages the previous history of the ODE trajectory as a more effective source of coarse-grained information. ARD modifies the teacher transformer architecture by adding token-wise time embedding to mark each input from the trajectory history and employs a block-wise causal attention mask for training. Furthermore, incorporating historical inputs only in lower transformer layers enhances performance and efficiency. We validate the effectiveness of ARD in a class-conditioned generation on ImageNet and T2I synthesis. Our model achieves a $5\times$ reduction in FID degradation compared to the baseline methods while requiring only 1.1% extra FLOPs on ImageNet-256. Moreover, ARD reaches FID of 1.84 on ImageNet-256 in merely 4 steps and outperforms the publicly available 1024p text-to-image distilled models in prompt adherence score with a minimal drop in FID compared to the teacher. Project page: https://github.com/alsdudrla10/ARD.*

*Work done during an internship at Meta GenAI. †Meta GenAI. §ETH Zürich. ‡KAIST. Correspondence to: alsdudrla10@kaist.ac.kr




(a) Step Distillation (baseline)

(b) AutoRegressive Distillation (ours)

(c) A Comparison of distillation methods.

(d) A comparison of recent generative models.

Figure 2. (a, b) Overall scheme of the baseline and proposed distillation methods. The training trajectory is given by the teacher ODE. (c, d) Comparison of the efficiency-performance trade-offs of the distillation methods and public generative models on ImageNet 256p.

## 1. Introduction

Diffusion models currently dominate image synthesis landscape due to their striking generalization capabilities and unprecedented visual quality [8, 12, 56, 61]. Unlike generative adversarial networks (GANs) [16], the stable training of DMs facilitates their expansion to high-resolution image generation. Recently, models based on Diffusion Transformers (DiT) [54] architecture gained significant popularity due to their excellent scaling properties and ability to generate high-resolution images [5, 6]. However, sampling from DMs requires repeated neural network evaluations [44], which makes the high-resolution image synthesis slow and resource-intensive.

DMs generate samples by solving the denoising process numerically. The denoising process has a probability flow ordinary differential equation (ODE) formulation [69, 71], which provides deterministic coupling between noise and samples. To reduce sampling costs, a series of distillation models [17, 27, 42, 45, 62, 72, 86] have been developed that learn to predict ODE solution with fewer steps. However, few-step student models suffer from exposure bias [53, 57] because the student's intermediate prediction often deviates from the teacher's ODE due to estimation errors. The errors accumulate during iterative sampling, causing the prediction to become more erroneous as we approach a solution.

To address exposure bias in few-step distillation models, we propose an AutoRegressive Distillation (ARD) method for diffusion transformers. ARD predicts the next sample $\mathbf{x}_{\tau_{s-1}}$ based on both the current estimate $\mathbf{x}_{\tau_s}$ and the entire historical trajectory, which is more informative. This approach offers two benefits: it reduces accumulated errors and provides a better source of coarse-grained information which is contained in the historical trajectory. Incorporating the historical trajectory in the lower layers further introduces an inductive bias to handle coarse-grained information. We find that when distilling based on the whole historical trajectory, the FID degradation from the teacher is five times lower than that of the baselines on ImageNet 256p, with only 1.1% more computation required. Our approach also scales well and can be used to distill 1024p text-to-image diffusion transformers, which outperform public distillation approaches in text-image alignment metrics.

## 2. Preliminary

### 2.1. Diffusion models

Diffusion models define a forward process and a corresponding reverse process with Stochastic Differential Equations (SDEs). The forward process in Eq. (1) maps from the data $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$ to a noise $\mathbf{x}_T$.

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}_t, \tag{1}$$

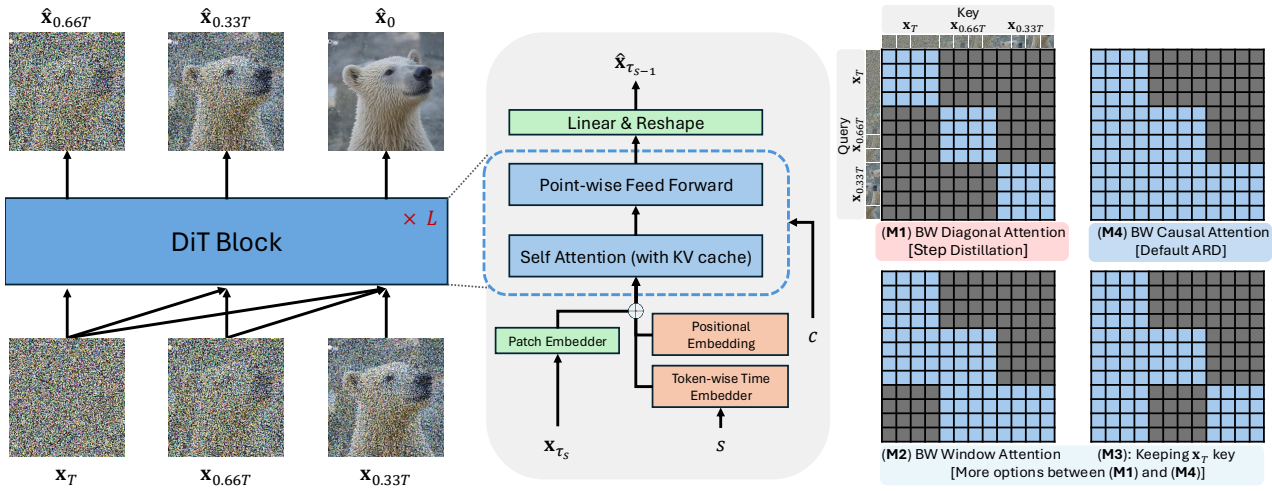
where $\mathbf{f} : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is a drift term, $g : [0, T] \to \mathbb{R}$ is a diffusion term, and $\mathbf{w}_t$ is a Wiener process. The forward process is often set to variance-preserving [22] or variance-exploding [71] SDEs to closely resemble a Gaussian distribution at $t = T$. Diffusion models generate the data from the noise $\mathbf{x}_T \sim p_{\text{prior}}(\mathbf{x}_T)$ through a reverse process [1, 71]. There exists a probability flow ODE (PF-ODE), which is a deterministic counterpart of the reverse process:

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)]dt. \tag{2}$$

Here $p_t(\mathbf{x}_t)$ is the marginal distribution defined by the forward process in Eq. (1). PF-ODE has the same marginal distribution as the reverse SDE while providing deterministic coupling between the noise $\mathbf{x}_T$ and the sample $\mathbf{x}_0$. Since the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is intractable, it is estimated by a neural network $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p_t^{\boldsymbol{\phi}}(\mathbf{x}_t)$ with a score matching objective [70, 79].

### 2.2. Step distillation models

The solution of an ODE in Eq. (2) is obtained by $\mathbf{x}_T + \int_T^0 \frac{d\mathbf{x}_t}{dt} dt$; however, it requires a sufficient number of steps to reduce discretization error [9, 44]. In order to compute $\frac{d\mathbf{x}_t}{dt}$ at each step, we need to evaluate the learned neural score function $\nabla_{\mathbf{x}_t} \log p_t^{\boldsymbol{\phi}^*}(\mathbf{x}_t)$, leading to high computational costs. To make inference efficient, step distillation [51, 62] defines intermediate times $\tau_s := T \times \frac{s}{S}$




(a) A transformer architecture of ARD. The DiT parameters are shared across each input $\mathbf{x}_{\tau_s}$. (b) Attention mask options. (BW: Block-Wise)

Figure 3. (a) The proposed transformer architecture for ARD. (b) The visualization of generalized mask options used during training: M1 represents step distillation, while M4 is the default setting of ARD. M2 and M3 are intermediate options between M1 and M4.

with $S$ as the total number of student steps and $s \in \{0, 1, \ldots, S\}$. These intermediate times define a trajectory $\boldsymbol{\mu}_{\phi^*} := [\mathbf{x}_{\tau_S}, \mathbf{x}_{\tau_{S-1}}, \ldots, \mathbf{x}_{\tau_1}, \mathbf{x}_{\tau_0}]$ within the teacher ODE starting from an initial noise $\mathbf{x}_{\tau_S} = \mathbf{x}_T$ and ending with a clean sample $\mathbf{x}_{\tau_0} = \mathbf{x}_0$. The student model learns a joint probability $p(\boldsymbol{\mu}_{\phi^*})$ defined as:

$$p(\boldsymbol{\mu}_{\phi^*}) = p_{\text{prior}}(\mathbf{x}_{\tau_S}) \times \prod_{s=1}^{S} p(\mathbf{x}_{\tau_{s-1}} | \mathbf{x}_{\tau_s}) \tag{3}$$

By the deterministic nature of PF-ODE, each conditional probability $p(\mathbf{x}_{\tau_{s-1}} | \mathbf{x}_{\tau_s})$ is a Dirac delta distribution, so it can be modeled by the deterministic mapping function; $\mathbf{x}_{\tau_{s-1}} = G(\mathbf{x}_{\tau_s}, s) := \mathbf{x}_{\tau_s} + \int_{\tau_s}^{\tau_{s-1}} \frac{d\mathbf{x}_t}{dt} dt$. The student model $G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_s}, s) \approx G(\mathbf{x}_{\tau_s}, s)$ learns to mimic the ground truth ODE integrations. Progressive distillation [51, 62] proposes a progressive algorithm for step distillation. However, such algorithm suffers from a significant drawback: the accumulation of errors during its iterative training phases when the student becomes the teacher again. Training a few-step student model directly from the teacher using $\mathcal{L}_{\text{step}}$ mitigates the accumulated errors brought the iterative progressive distillation procedure. We build our method on top of step distillation, where we directly learning from the teacher:

$$\mathcal{L}_{\text{step}} := \mathbb{E}_{\boldsymbol{\mu}_{\phi^*}} \left[ \sum_{s=1}^{S} ||G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_s}, s) - \mathbf{x}_{\tau_{s-1}}||_2^2 \right]. \tag{4}$$

**Exposure bias** During inference, the generation starts from $\mathbf{x}_{\tau_S} \sim p_{\text{prior}}(\mathbf{x}_{\tau_S})$. At each step, the student model predicts $\hat{\mathbf{x}}_{\tau_{s-1}} = G_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{\tau_s}, s)$ based only on the current sample $\hat{\mathbf{x}}_{\tau_s}$. If $\hat{\mathbf{x}}_{\tau_s}$ deviates from the teacher ODE, the student model $G_{\boldsymbol{\theta}}$ infers based on an unseen sample that was not encountered during training. Consider, for example, the intermediate samples depicted in Fig. 2a, where a fish is shown without eyes, despite such samples did not appear in the training data. This unforeseen input propagates through the sampling process, culminating in a final sample $\mathbf{x}_{\tau_0}$ that also lacks eyes. This exposure bias is an inherent limitation of the iterative procedure [53, 57], unless perfect optimization is achieved. The errors accumulate as the iterative sampling process progresses.

### 2.3. Autoregressive models

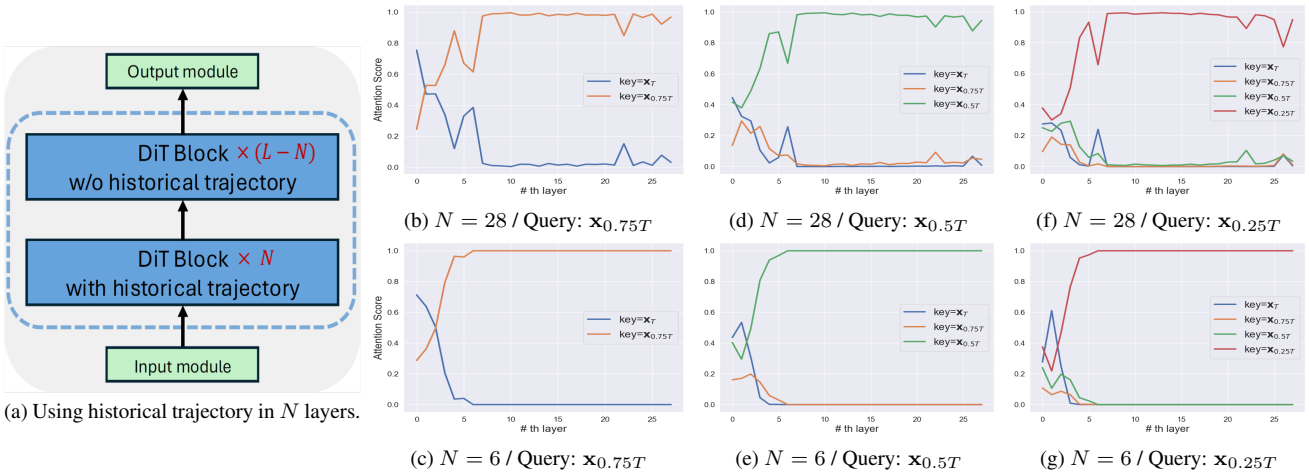
Autoregressive models [32] represent the joint probability distribution of a multivariate random variable $\mathbf{x} := [x_S, x_{S-1}, \ldots, x_0]$ by decomposing it into a product of conditional probabilities $p(\mathbf{x}) = p(x_S) \times \prod_{s=1}^{S} p(x_{s-1} | \mathbf{x}_{S:s})$, where $\mathbf{x}_{S:s} = [x_S, x_{S-1}, \ldots, x_s]$. This formulation, as depicted above, does not rely on any specific assumptions. Each component $p(x_{s-1} | \mathbf{x}_{S:s})$ of the decomposition incorporates the information of all preceding variables.

## 3. Method

In this section we introduce the AutoRegressive Distillation (ARD) of diffusion transformers (DiT). Figure 2b provides an overview of the ARD process. We'll break down the probabilistic formulations of distillation in Section 3.1, then move on to the transformer architecture design for our student model in Section 3.2. Lastly, we'll cover training and inference in Section 3.3.

### 3.1. Autoregressive distillation

This section generalizes the step distillation formulation in Eq. (3) to ARD. The decomposition in Eq. (3) is valid without whole historical trajectory information under perfect




(a) Using historical trajectory in $N$ layers.

(b) $N = 28$ / Query: $\mathbf{x}_{0.75T}$ (d) $N = 28$ / Query: $\mathbf{x}_{0.5T}$ (f) $N = 28$ / Query: $\mathbf{x}_{0.25T}$

(c) $N = 6$ / Query: $\mathbf{x}_{0.75T}$ (e) $N = 6$ / Query: $\mathbf{x}_{0.5T}$ (g) $N = 6$ / Query: $\mathbf{x}_{0.25T}$

Figure 4. (a) shows an additional inductive bias that we impose by using the historical trajectory in lower layers only. (b, d, f) show the attention scores for each history input (key tokens) during the 2nd, 3rd, 4th steps when $N = L$. (c, e, g) show the same but with $N = 6$. The attention score on input $\mathbf{x}_{\tau_s}$, is the sum of attention weights for all key tokens in $\mathbf{x}_{\tau_{s'}}$, indicating the portion of $\mathbf{x}_{\tau_{s'}}$.

distillation. However, when each probability $p(\mathbf{x}_{\tau_{s-1}}|\mathbf{x}_{\tau_s})$ is approximated by $\hat{\mathbf{x}}_{\tau_{s-1}} = G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_s}, s)$, the discrepancy with the ground truth is inevitable due to estimation error, leading to the exposure bias problem discussed in Sec. 2.2.

To mitigate this problem, we extend the formulation of Eq. (3) in an autoregressive manner motivated by Sec. 2.3:

$$p(\boldsymbol{\mu}_{\boldsymbol{\phi}^*}) = p_{\text{prior}}(\mathbf{x}_{\tau_S}) \times \prod_{s=1}^{S} p(\mathbf{x}_{\tau_{s-1}}|\mathbf{x}_{\tau_S:\tau_s}), \qquad (5)$$

where $\mathbf{x}_{\tau_S:\tau_s} = [\mathbf{x}_{\tau_S}, \mathbf{x}_{\tau_{S-1}}, \dots, \mathbf{x}_{\tau_s}]$ denotes the historical trajectory. This formulation has two benefits: (i) Every step includes the ground truth initial noise $\mathbf{x}_{\tau_S}$ as input, which has a deterministic coupling with the prediction target $\mathbf{x}_{\tau_{s-1}}$. Furthermore, the historical trajectory predictions from $\hat{\mathbf{x}}_{\tau_{S-1}}$ to $\hat{\mathbf{x}}_{\tau_{s+1}}$ are more accurate compared to the recent sample $\hat{\mathbf{x}}_{\tau_s}$ because for them the error had less chances to accumulate during inference. In contrast, the input in Eq. (3) is merely the current sample $\hat{\mathbf{x}}_{\tau_s}$, making it vulnerable to exposure bias. (ii) To predict $\mathbf{x}_{\tau_{s-1}}$ at every step, the model needs to generate both coarse-grained and fine-grained information. The recent denoised sample $\mathbf{x}_{\tau_s}$ is the best source for fine-grained information, but the historical trajectory close to $\mathbf{x}_{\tau_S}$ is a better source for coarse-grained information [13, 60].

For the modified student formulation, we aim to estimate $p(\mathbf{x}_{\tau_{s-1}}|\mathbf{x}_{\tau_S:\tau_s})$, which is still a Dirac delta distribution. To achieve this we define a new mapping function $\mathbf{x}_{\tau_{s-1}} = G(\mathbf{x}_{\tau_S:\tau_s}, s) := \mathbf{x}_{\tau_s} + \int_{\tau_s}^{\tau_{s-1}} \frac{d\mathbf{x}_t}{dt} dt$. This function is then approximated by a student neural network $G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_S:\tau_s}, s)$.

### 3.2. Transformer design

The design of our mapping function $G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_S:\tau_s}, s)$ defined in Sec. 3.1 is not trivial because the input size varies depending on the denoising step $s$. To overcome this, we modify the teacher DiT backbone to accommodate multiple inputs.

**Architecture** To handle the historical trajectory, we design transformer-based autoregressive model as shown in Fig. 3a. Each input $\mathbf{x}_{\tau_s}$ is tokenized into a sequence of tokens using a shared patch embedder. Since each input $\mathbf{x}_{\tau_s}$ has the same spatial structure as a 2D grid, positional embeddings are shared across the inputs. The transformer blocks need to identify the order of each token in the inputs sequence $\mathbf{x}_{\tau_S}, \dots, \mathbf{x}_{\tau_s}$. To this end, we add an extra time-step embeddings to each token similar to the level embedding in VAR [76].[1] The recent denoised sample $\mathbf{x}_{\tau_s}$ becomes the query tokens, and the history sequence $\mathbf{x}_{\tau_S:\tau_s}$ becomes the key-value tokens in the self-attention blocks. After passing through $L$ stacked transformer blocks, the tokens are linearly transformed and de-tokenized to obtain a sample $\mathbf{x}_{\tau_{s-1}}$.

**Historical trajectory only in lower $N$ layers.** Figures 4b, 4d and 4f show the attention scores of each input in (2nd, 3rd, 4th) steps at each $L$ transformer layers. The recent denoised sample $\mathbf{x}_{\tau_s}$ is most activated as key tokens in the higher layers, while the historical trajectory $\mathbf{x}_{\tau_S:\tau_{s+1}}$ is activated in the lower layers. The lower layers in DiT blocks are known to consider coarse-grained information, while the higher layers in DiT blocks are considered fine-grained information [19]. This attention portion validates that the historical trajectory is useful and serves as a better source of coarse-grained information. However, the his-

[1] The original DiT backbone uses time embedding with adaLN [55] because the tokens in the DiT teacher are always from the same input, so it does not need to identify time on a token-wise basis. On the other hand, our student model needs to be modified to identify the origin of each token.

torical tokens still slightly fluctuate in the higher layers in Figs. 4b, 4d and 4f, possibly due to imperfect optimization. We propose additional design choices in transformer layers as shown in Fig. 4a; using the historical trajectory only in the lower $N$ layers. This inductive bias enhances the use of the historical trajectory in the lower layers as shown in Figs. 4c, 4e and 4g.

### 3.3. Training and inference procedure

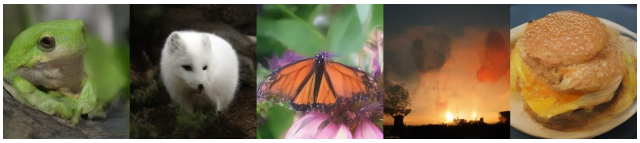
The default training objective of ARD is a regression loss $\mathcal{L}_{\text{ARD}}$ in Eq. (6), and it is optimized with respect to $\boldsymbol{\theta}$. The transformer architecture in Fig. 3a allows computing $\hat{\mathbf{x}}_{\tau_{s-1}} = G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_S:\tau_s}, s)$ for all $s \in \{1, \dots, S\}$ simultaneously by using an attention mask. We can generalize our framework by designing the attention mask as shown in Fig. 3b. Block-wise causal attention in option M4 is the most flexible, as it uses the entire trajectory history. Option M1 represents step distillation, which only uses the current sample $\mathbf{x}_{\tau_s}$ as input. Options M2 and M3 are intermediate choices between M1 and M4. The windowed attention in M2 uses only the current and the previous sample from the trajectory history. The attention mask in M3 uses the most recent denoised sample and the initial noise $\mathbf{x}_{\tau_S}$, which helps to consistently preserve the ground truth signal. Our framework can also benefit from an additional discriminator loss for the final prediction $\hat{\mathbf{x}}_{\tau_0} = G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_S:\tau_1}, 1)$, similar to [27]. By using real data in this loss, we can further improve the high-frequency details in the student generations, even outperforming the teacher.

$$\mathcal{L}_{\text{ARD}} := \mathbb{E}_{\boldsymbol{\mu}_{\boldsymbol{\phi}^*}} \left[ \sum_{s=1}^{S} ||G_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_S:\tau_s}, s) - \mathbf{x}_{\tau_{s-1}}||_2^2 \right] \tag{6}$$

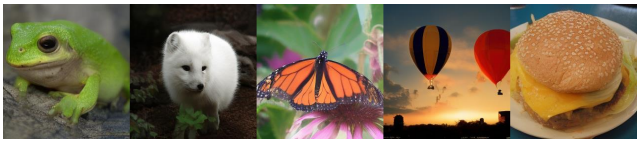
During inference, the generation starts from $\mathbf{x}_{\tau_S} \sim p_{\text{prior}}(\mathbf{x}_{\tau_S})$. At each step, the student model predicts $\hat{\mathbf{x}}_{\tau_{s-1}} = G_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{\tau_S:\tau_s}, s)$ based on the entire historical predictions $\hat{\mathbf{x}}_{\tau_S:\tau_s} = [\mathbf{x}_{\tau_S}, \hat{\mathbf{x}}_{\tau_{S-1}}, \dots, \hat{\mathbf{x}}_{\tau_s}]$. The information of $[\mathbf{x}_{\tau_S}, \hat{\mathbf{x}}_{\tau_{S-1}}, \dots, \hat{\mathbf{x}}_{\tau_{s+1}}]$ is stored as kv-cache in the pre-

Table 1. Comparison with distillation methods from the same teacher. The teacher uses DiT-XL/2 architecture trained on ImageNet 256p. Loss R denotes the use of regression loss for distillation, and R+D denotes using additional discriminator loss. The FLOPs and latency are measured during the denoising process. Latency refers to the time required to generate one image measured on an H100.

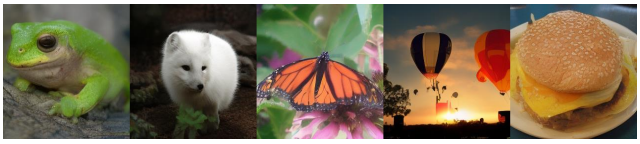
| Model | Loss | Mask | Steps ($S$) ↓ | GFLOPs ↓ | Latency (ms) ↓ | FID ↓ | IS ↑ | Prec ↑ | Rec ↑ |
|---|---|---|---|---|---|---|---|---|---|
| DiT/XL-2 | - | - | 250 | 59300 | 4935 | 2.27 | 278.24 | 0.830 | 0.570 |
| (target teacher) | - | - | 25 | 5930 | 493.5 | 2.89 | 230.22 | 0.797 | 0.572 |
| KD [45] | R | - | 1 | 118.6 | 17.01 | 11.88 | 148.61 | 0.665 | 0.565 |
| Step Distill. [62] ($N = 0$) | R | M1 | 2 | 237.2 | 33.05 | 10.92 | 167.08 | 0.681 | 0.518 |
| ARD ($N = 6$) | R | M4 | 2 | 238.1 | 34.05 | **6.29** | **188.05** | **0.737** | 0.564 |
| ARD ($N = 28$) | R | M4 | 2 | 241.5 | 34.90 | 6.54 | 186.18 | 0.734 | **0.569** |
| Step Distill. [62] ($N = 0$) | R | M1 | 4 | 474.4 | 64.80 | 10.25 | 181.58 | 0.704 | 0.474 |
| ARD ($N = 6$) | R | M2 | 4 | 477.1 | 65.57 | 4.75 | 203.58 | 0.768 | 0.572 |
| ARD ($N = 6$) | R | M3 | 4 | 477.1 | 65.57 | 4.45 | 206.93 | 0.773 | 0.572 |
| ARD ($N = 6$) | R | M4 | 4 | 479.9 | 66.34 | **4.32** | **209.03** | **0.770** | **0.574** |
| ARD ($N = 28$) | R | M4 | 4 | 500.2 | 67.85 | 4.80 | 201.15 | 0.761 | 0.566 |
| Step Distill. [62] ($N = 0$) | R+D | M1 | 4 | 474.4 | 64.80 | 3.84 | 221.16 | 0.785 | 0.557 |
| ARD ($N = 6$) | R+D | M4 | 4 | 479.9 | 66.34 | **1.84** | **235.84** | **0.797** | **0.615** |



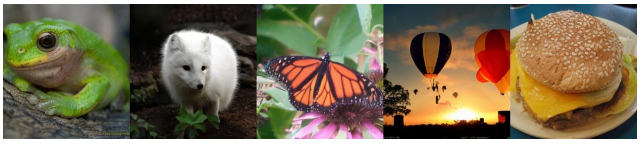
(a) Step Distillation (R) / FID: 10.25



(c) Teacher (25 steps) / FID: 2.89

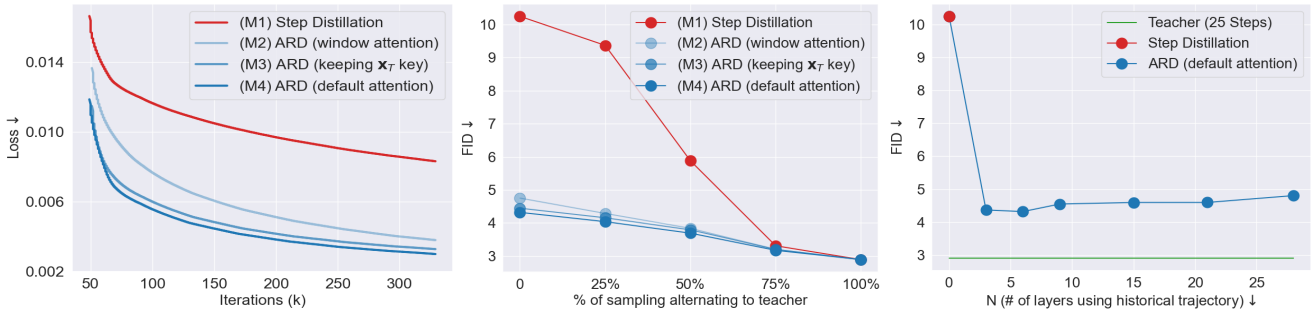


(b) ARD (R) / FID: 4.32



(d) ARD (R+D) / FID: 1.84

Figure 5. Generated ImageNet 256p samples from same initial noise $\mathbf{x}_{\tau_S}$. All distilled models are 4-step models.




(a) Training curve on 4-step models. (b) Error in each step in 4-step models (c) $N$ ablations on 4-step models.

Figure 6. The analysis on design choices (attention mask options & $N$) for 4-step distillation methods.

vious steps for fast inference. No attention mask is required during inference.

## 4. Experiments

This section empirically validates the effectiveness of ARD. Section 4.1 explains the results of class-conditional image synthesis on ImageNet [10]. Section 4.2 presents the experimental results for text-conditional image synthesis.

### 4.1. Class-conditional image generation

We use a DiT/XL-2 latent diffusion transformer architecture following [54], and employ it as a teacher architecture. The teacher ($\phi$) is trained on ImageNet 256p. We construct an ODE trajectory $\boldsymbol{\mu}_{\phi^*}$ by running the teacher with 25 steps and a classifier-free guidance scale [21] of 1.5. In total, we pre-compute and store 2.56M ODE trajectories for the distillation.

**Evaluation metrics** To evaluate sample fidelity and diversity, we measure FID [20], IS [63], Precision, and Recall [31] following the protocol of ADM [12] with a pre-trained Inception-V3 Network [75]. FID and IS quantify both sample quality and diversity. Precision measures sample fidelity, while Recall measures sample diversity by quantifying the manifold overlap region between real and generated samples in the feature space.

**Performance gain from ARD** The proposed ARD is a generalization over previous methods [45, 62]: When the number of steps $S$ is 1, ARD becomes Knowledge Distillation (KD) [45]. When ARD is used with attention mask option M1 in Fig. 3b, it becomes step distillation [62]. Table 1 and Fig. 2c show the performance gain from the extended design of ARD. Increasing the number of steps from 2 to 4 for our method results in better performance in FID: $6.29 \rightarrow 4.32$. However, the FID gain in step distillation is marginal ($10.92 \rightarrow 10.25$) even when the number of steps $S$ increases from 2 to 4, moreover Recall decreases significantly. In Figures 5a and 5c we can see that step distillation fails to preserve the global structure (e.g., the orientation of a frog), indicating that the diversity of the teacher's solution is not maintained. This happens because as the number of steps increases in step distillation, it becomes harder to preserve the deterministic coupling between the initial noise $\mathbf{x}_{\tau_S}$ and the sample $\mathbf{x}_{\tau_0}$ provided by the teacher due to increased exposure bias (see Section 2.2).

Table 2. Comparison with public models on ImageNet 256p

| Type | Model | Params ↓ | Steps ↓ | FID ↓ | Rec ↑ |
|---|---|---|---|---|---|
| GAN | BigGAN [3] | 112M | 1 | 6.95 | 0.38 |
| GAN | StyleGAN-XL [65] | 166M | 1 | 2.30 | 0.53 |
| GAN | GigaGAN [24] | 569M | 1 | 3.45 | 0.61 |
| DM | ADM [12] | 554M | 250 | 4.59 | 0.52 |
| DM | LDM [61] | 400M | 250 | 3.60 | 0.48 |
| DM | DiT [54] | 675M | 250 | 2.27 | 0.57 |
| DM | VDM++ [29] | 2.0B | 250 | 2.40 | - |
| NAT | MaskGIT [4] | 227M | 8 | 6.18 | 0.51 |
| NAT | RCG [37] | 502M | 20 | 2.15 | 0.53 |
| NAT | AutoNAT [52] | 194M | 8 | 2.68 | - |
| AR | VQVAE2 [59] | 13.5B | 5120 | 31.11 | 0.57 |
| AR | VQGAN [15] | 227M | 256 | 18.65 | 0.26 |
| AR | RQTran. [33] | 3.8B | 68 | 7.55 | - |
| AR | VAR [76] | 2.0B | 10 | 1.97 | 0.59 |
| AR | MAR [38] | 943M | 32 | 1.93 | - |
| AR | ARD (Ours) | 675M | 4 | **1.84** | **0.62** |

ARD with a block-wise causal mask M4 (using the entire trajectory history) outperforms step distillation in all metrics for 2 and 4 steps. Unlike step distillation, ARD maintains its Recall as the number of steps increases from 2 to 4. Figures 5b and 5c show that ARD preserves the global structure of the teacher's solution. As a result, ARD improves significantly as the number of steps $S$ increases. Figure 6a shows that ARD variants (Eq. (6)) converge more effectively than vanilla regression loss (Eq. (4)) during training, demonstrating the benefits of our autoregressive design. Moreover, for a 4-step ARD model the FID degradation from the teacher is $1.43 = (4.32 - 2.89)$, which is $5\times$ lower than that of step distillation $7.36 = (10.25 - 2.89)$.

When using additional discriminator loss (R+D), the performance of both step distillation and ARD improves, with ARD outperforming step distillation and achieving FID of

1.84. Figure 5d shows that discriminator loss makes the samples sharper while maintaining the global structure from the coupling $[\mathbf{x}_{\tau_S}, \mathbf{x}_{\tau_0}]$ provided by the teacher. This indicates the additional discriminator loss does not harm the diversity of the samples, and even improves the Recall metrics. The discriminator loss enables ARD to perform better than the teacher, and outperform the public few-step generative models in Table 2 and Fig. 2d in the low-step regime both in speed and quality. The speed in Fig. 2d was measured on an NVIDIA H100 with a batch size of 128. For the baselines, we used their official code and measured the speed under the same conditions.

**Ablation of the attention mask** Table 1 shows the results for various attention masks introduced in Fig. 3b when $S = 4$ and $N = 6$. The default attention mask M4 exhibits the best FID of 4.32 due to its flexibility. The window attention in M2 and the retention of initial noise in M3 achieve FIDs of 4.75 and 4.45, respectively. These relaxed options, M2 and M3, also show significant gains over step distillation (10.25). Note that both M2 and M3 use two inputs at each step. M2 sets the window size to 2, so two recent denoised samples $[\mathbf{x}_{\tau_{s+1}}, \mathbf{x}_{\tau_s}]$ are used to predict $\mathbf{x}_{\tau_{s-1}}$. M3 uses $[\mathbf{x}_{\tau_S}, \mathbf{x}_{\tau_s}]$ as inputs. Using the initial noise $\mathbf{x}_{\tau_S}$ (M3) appears more beneficial as additional information, as it helps maintain a ground truth input signal at every step. The analysis (in Figures 4c, 4e and 4g) of the attention scores with block-wise causal mask M4 show that the initial noise is the most activated among $\mathbf{x}_{\tau_S:\tau_{s+1}}$, demonstrating the importance of information in it.

**Error accumulation ablation** When we sample with a student model starting from ground truth points on the teacher's trajectory, we can analyze which steps accumulate more errors in the student models. As shown in Fig. 6b, if the first three steps are solved by the teacher (75%), the performance gap between step distillation and ARD is small. However, if early steps are predicted by the student, the performance of step distillation drops significantly. This suggests that step distillation is more susceptible to exposure bias, whereas ARD is more robust.

**Ablation of $N$** For 4-step ARD model, optimal performance is achieved at $N = 6$, as shown in Fig. 6c. While a large $N$ makes the student model flexible, a small $N$ provides an effective inductive bias. $N = 6$ performs well in 2-step cases (see Table 1) and with different attention masks too. For window attention M2, FID improves from 5.08 to 4.75. For M3, which keeps the initial noise, FID improves from 5.01 to 4.45 as $N$ decreases from 28 to 6. Additionally, smaller $N$ results in faster inference and requires less memory as we need to store kv-cache for fewer layers.

**Efficiency analysis** Table 1 and Fig. 2c show the floating-point operations (FLOPs) in the denoising process during the inference phase, indicating the theoretical computational costs. The inference FLOPs for the backbone DiT architecture are 118.6 GFLOPs [54]. The FLOPs for step distillation are proportional to the number of steps $S$, and the teacher requires twice the FLOPs due to classifier-free

Table 3. Text-image alignment scores on CompBench for public high-resolution ($\geq$ 768p) distillation models. Loss R denotes the use of regression loss, and D denotes the use of discriminator loss. The best and second-best results are highlighted in **bold** and underline.

| Model | Params ↓ | Steps ↓ | Res. | CFG | Loss | CompBench ↑ (%) Color | Shape | Texture | Spatial | Non-spatial | Complex | AVG ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Step distill. [62] | 2.7B | 4 | 768 | 6 | R | 44.1 | 26.9 | 40.0 | 48.7 | 54.3 | 39.6 | 42.3 |
| ADD [68] | 2.7B | 3 | 768 | 6 | D | 43.3 | 32.9 | 44.5 | 42.6 | 56.2 | 36.5 | 42.6 |
| Imagine Flash [30] | 2.7B | 3 | 768 | 6 | R+D | 42.7 | 36.2 | 47.9 | 57.4 | 62.9 | 42.8 | 48.3 |
| Lightning [39] | 2.6B | 4 | 1024 | 6 | D | 57.1 | 46.5 | 53.1 | 62.0 | 61.4 | 41.9 | 53.7 |
| DMD2 [82] | 2.6B | 4 | 1024 | 8 | R+D | 64.5 | 47.8 | 60.4 | **68.9** | **66.3** | **49.1** | 59.5 |
| Pixart-delta [7] | 0.6B | 3 | 1024 | 4.5 | R | 38.7 | 33.2 | 40.8 | 56.1 | 60.1 | 39.5 | 44.7 |
| LCM-LoRA [47] | 1.4B | 3 | 1024 | 7.5 | R | 49.3 | 35.2 | 45.0 | 54.5 | 57.0 | 39.9 | 46.8 |
| LCM-LoRA [47] | 2.8B | 3 | 1024 | 7.5 | R | 59.5 | 45.3 | 50.2 | 55.3 | 60.8 | 44.6 | 52.6 |
| ARD | 1.7B | 3 | 1024 | 3 | R | 64.6 | 46.6 | 63.5 | 60.0 | 57.8 | 44.8 | 56.2 |
| ARD | 1.7B | 3 | 1024 | 7.5 | R | **71.1** | **53.1** | **65.7** | 64.4 | 61.2 | 44.0 | **59.9** |

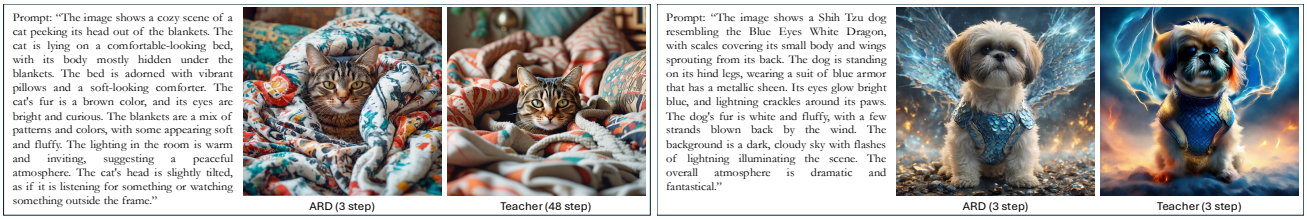


Figure 7. Sample comparison between ARD and the Emu teacher on long (realistic/unrealistic) prompts.

guidance [21]. ARD requires slightly more FLOPs due to the kv-cache, which increases the number of attended keys and the aggregation of the respective values. The increased amount is proportional to $N$, which is the number of layers using the attention cache. ARD ($N = 6$), which is our best model, only uses 1.1% more FLOPs compared to the step distillation models. Latency shows a similar trend to FLOPs, except for the teacher. The attention options M2 and M3 in Fig. 3b require slightly fewer FLOPs, at 477.1 GFLOPs in the 4-step cases, because the amount of kv-cache is smaller compared to the default attention. These relaxed options can significantly reduce the increase in kv-cache when S becomes larger.

### 4.2. Text-conditional image generation

We use 1.7B Emu [8] model with diffusion transformer architecture as a teacher. The teacher model was pre-trained for 1024p resolution on a large-scale internal dataset and fine-tuned on a small set of high-quality aesthetic images. We calculate the teacher ODE trajectories online and use 48 teacher steps by default. Since our distillation requires only the text prompt for training, we use a large-scale internal pre-training dataset for the distillation. For ARD student transformer we opt for a block-wise causal attention M4. We do 15k training iterations using only regression loss.

**Evaluation metrics** To evaluate sample fidelity and diversity, we compute zero-shot FID on the MS-COCO 2017 dataset [40] using 5k random real samples and generated samples from 5k random prompts. To measure prompt alignment, we use CompBench [23], which has six categories of prompts following the evaluation protocol of Imagine Flash [30].

**Text-Image alignment** Table 3 shows the text-image alignment scores on CompBench compared to public high-res ($\geq$768 pix) distilled models. Our ARD based on Emu (7.5 CFG) outperforms all other public high-res distilled models in the average score. ARD surpasses all other 1024p distillation models in all six categories except DMD2 [82] which shows comparable performance, but uses 0.9B more parameters and more sampling steps. In ARD distilled from Emu (3.0 CFG), the average score gap between the student and the teacher is only 2.3, which is the smallest among all 3-step distillation models (see supplementary for each teacher's performance). ARD can also successfully generate images following long and detailed prompts (see Figure 7).

**Sample quality** Table 4 shows the FID comparison across 1024p public distillation methods, which target the teacher's PF-ODE. While the absolute performance of the student (FID-S) is the second best among the models, the best model LCM-LoRA (2.8B) has 1.1B more parameters than ARD. Since the upper-bound performance of the distilled model is the teacher when the distillation targets PF-ODE, its performance highly depends on the teacher's performance (FID-T). Table 4 demonstrates a clear trend: teacher models with larger parameters achieve better performance in FID-T. To quantify the effectiveness of the distillation method, we measure the performance drop, which is the gap between the teacher and the student. ARD exhibits the smallest drop compared to the baselines. We trained step distillation from the same Emu teacher, and ARD still showed better performance, which further validates the benefit of using the trajectory history.

Table 4. Zero-shot FID 5k on MS-COCO 2017 for the public 1024p distillation models. FID-T is the teacher performance, and FID-S is the student performance. The drop indicates the performance gap between the students and the teachers.

| Model | Params ↓ | Steps ↓ | FID-T | FID-S | Drop ↓ |
|---|---|---|---|---|---|
| Lightning [39] | 2.6B | 4 | 24.30 | 30.16 | 5.86 |
| LCM-LoRA [47] | 2.8B | 3 | 25.11 | **27.77** | 2.66 |
| LCM-LoRA [47] | 1.4B | 3 | 30.79 | 33.79 | 3.00 |
| Step Distil. [51] | 1.7B | 3 | 27.97 | 30.51 | <u>2.54</u> |
| ARD | 1.7B | 3 | 27.97 | <u>30.03</u> | **2.06** |

Figure 1 shows the generated samples by ARD distilled from Emu (7.5 CFG). ARD generates high-quality images across various topics and styles. The left example in Fig. 7 compares samples from ARD and the target teacher with the same initial noise $\mathbf{x}_{\tau_S}$ and long detailed prompts with realistic contexts (left) and unrealistic contexts (right). The samples are not identical, but the image from ARD maintains high fidelity and effectively preserves text information. The detailed description of the subject and background is well captured in the images. The right example in Fig. 7 compares samples generated by both teacher and student in 3 steps. Although the number of steps is the same, ARD uses only extra kv-cache, whereas the teacher requires twice more computations due to CFG.

## 5. Conclusion

This paper introduced a novel few-step distillation method for the diffusion transformers that generalizes step distillation by leveraging the entire historical ODE trajectory in an autoregressive way. We also introduced a modified transformer architecture to support the autoregressive distillation design. The use of ODE trajectory mitigates exposure bias by maintaining the ground truth input signal at every step. By analyzing the historical trajectory as a better source of coarse-grained information, ARD introduces an additional design choice of using historical trajectory only in lower layers based on attention weight analysis. The empirical results show that ARD outperforms step distillation and surpasses public few-step generative models.

## References


[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 2

[2] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023. 1

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 6

[4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 6, 1

[5] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 2

[6] Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-$\alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 5

[7] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024. 7, 5

[8] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 2, 8, 5

[9] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021. 2

[10] 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

[11] 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)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 1

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 6

[13] Sander Dieleman. Diffusion is spectral autoregression, 2024. 4

[14] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems*, 2022. 1

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 6, 1, 2

[16] Ian 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. 2

[17] Jiatao Gu, Chen Wang, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. Data-free distillation of diffusion models with bootstrapping. In *Forty-first International Conference on Machine Learning*, 2024. 2, 1

[18] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly, Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable text-to-image generation, 2024. 1

[19] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *European Conference on Computer Vision*, pages 37–55. Springer, 2024. 4

[20] 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

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6, 8

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[23] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 8

[24] 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*, pages 10124–10134, 2023. 6

[25] Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling Diffusion Models into Conditional GANs. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 1

[27] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 5, 1

[28] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. PagoDA: Progressive growing of a one-step generator from a low-resolution diffusion teacher. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[29] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2023. 6

[30] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024. 7, 8, 1, 5

[31] 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

[32] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 29–37. JMLR Workshop and Conference Proceedings, 2011. 3

[33] 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. 6, 1

[34] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 18957–18973. PMLR, 2023. 1

[35] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[36] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis, 2023. 1

[37] Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6

[38] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6

[39] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 7, 8, 1

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 8

[41] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[42] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[43] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[44] 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. 2, 1

[45] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 2, 5, 6, 1

[46] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 1

[47] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. 7, 8, 5

[48] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2023. 1

[49] Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-Jun Qi. One-step diffusion distillation through score implicit matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[50] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 5

[51] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 2, 3, 8, 1

[52] Zanlin Ni, Yulin Wang, Renping Zhou, Jiayi Guo, Jinyi Hu, Zhiyuan Liu, Shiji Song, Yuan Yao, and Gao Huang. Revisiting non-autoregressive transformers for efficient image synthesis. In *Proceedings of the IEEE/CVF Conference*

*on Computer Vision and Pattern Recognition*, pages 7007–7016, 2024. 6, 1

[53] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[54] 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. 2, 6, 7

[55] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[56] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 5

[57] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016. 2, 3

[58] Rohit Rao. Announcing ssd-1b: A leap in efficient t2i generation, 2023. 5

[59] 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. 6, 1

[60] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[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, 6

[62] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 2, 3, 5, 6, 7, 1

[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] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021. 2

[65] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 6

[66] 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*, pages 30105–30118. PMLR, 2023. 2

[67] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation, 2024. 1, 2

[68] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024. 7, 1

[69] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 4

[70] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2

[71] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[72] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 2, 1

[73] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024. 1

[74] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[75] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[76] Keyu Tian, Yi Jiang, Zehuan Yuan, BINGYUE PENG, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4, 6, 1

[77] 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

[78] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 1

[79] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 2

[80] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang,

Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1

[81] Sirui Xie, Zhisheng Xiao, Diederik P Kingma, Tingbo Hou, Ying Nian Wu, Kevin Patrick Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. EM distillation for one-step diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[82] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 7, 8

[83] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6613–6623, 2024. 1

[84] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. Featured Certification. 1

[85] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[86] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, pages 42390–42402. PMLR, 2023. 2, 1