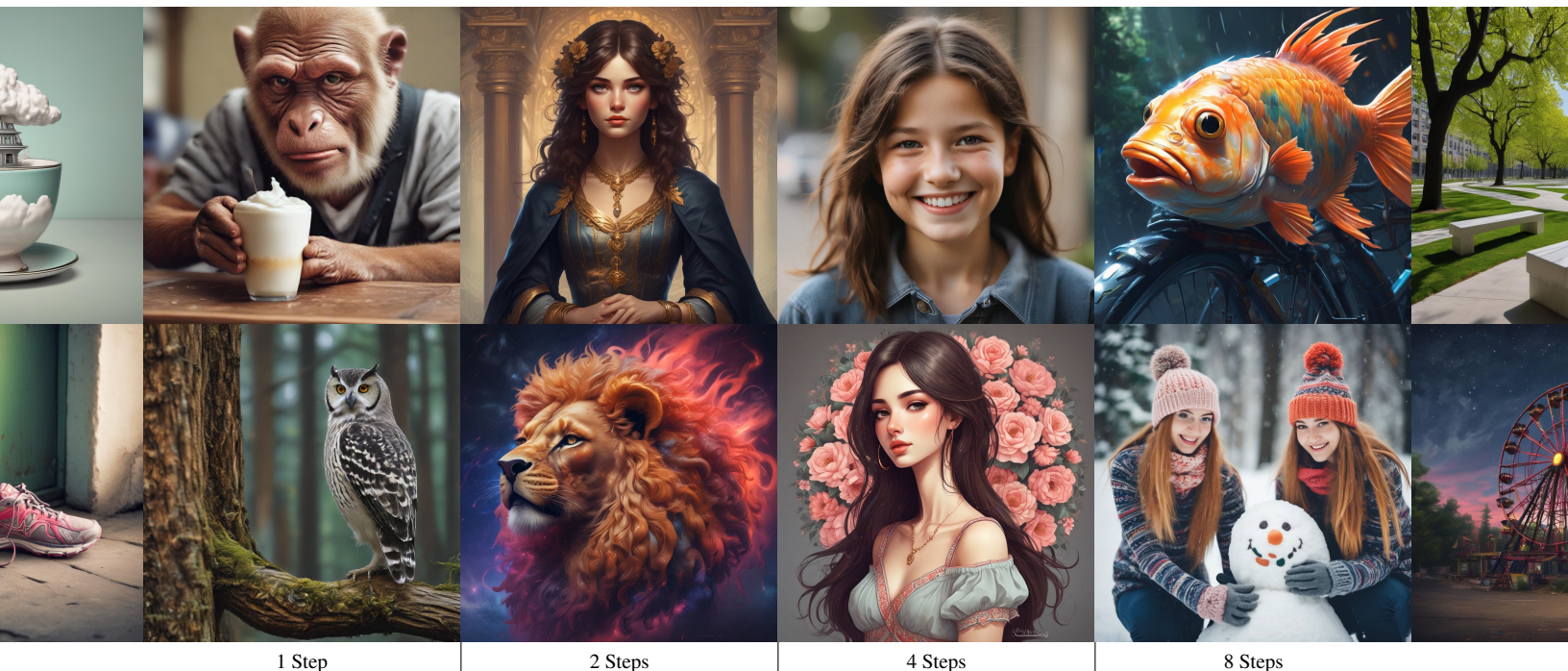# SDXL-Lightning: Progressive Adversarial Diffusion Distillation

Shanchuan Lin   Anran Wang   Xiao Yang
ByteDance Inc.
{peterlin, anran.wang, yangxiao.0}@bytedance.com

1 Step   2 Steps   4 Steps   8 Steps

Abstract

We propose a diffusion distillation method that achieves new state-of-the-art in one-step/few-step 1024px text-to-image generation based on SDXL. Our method combines progressive and adversarial distillation to achieve a balance between quality and mode coverage. In this paper, we discuss the theoretical analysis, discriminator design, model formulation, and training techniques. We open-source our distilled SDXL-Lightning models both as LoRA and full UNet weights.

Model: https://huggingface.co/ByteDance/SDXL-Lightning

1. Introduction

Diffusion models [14, 63, 67] are a rising class of generative models that has achieved state-of-the-art results in a wide range of applications, such as text-to-image [5, 43, 44, 49, 51, 53, 79], text-to-video [2, 3, 10, 13, 62, 80], and image-to-video [2], *etc*. However, the iterative generation process of diffusion models is slow and computationally expansive. How to generate high-quality samples faster is an actively researched area and is the main focus of our work.

Conceptually, the generation involves a flow that gradually transports samples between the data and the noise distribution. The diffusion model learns to predict the gradient at any location of this flow. The generation is simply transporting samples from the noise distribution to the data

distribution by following the predicted gradient through the flow. Because the flow is complex and curved, the generation must take a small step at a time. Formally, the flow is expressed as ordinary differential equations (ODEs) [67]. In practice, generating a high-quality data sample requires more than 50 inference steps.

Different approaches to reduce the number of inference steps have been researched. Prior works have proposed better ODE solvers to account for the curving nature of the flow [19, 30, 34, 35, 64, 78]. Others have proposed formulations to make the flow straighter [29, 31]. Nonetheless, these approaches generally still require more than 20 inference steps.

Model distillation [22,32,36,37,54,58,65,66,73,75], on the other hand, can achieve high-quality samples under 10 inference steps. Instead of predicting the gradient at the current flow location, it changes the model to directly predict the next flow location much farther ahead. Existing methods can achieve good results using 4 or 8 inference steps, but the quality is still not production-acceptable using 1 or 2 inference steps. Our method falls under the model distillation umbrella and achieves much superior quality compared to existing methods.

Our method combines the best of both worlds from progressive [54] and adversarial distillation [58]. Progressive distillation ensures that the distilled model follows the same ODE flow and has the same mode coverage as the original model. However, progressive distillation with mean squared error (MSE) loss produces blurry results under 8 inference steps and we provide theoretical analysis in our paper. To mitigate the issue, we use adversarial loss at every stage of the distillation to strike a balance between quality and mode coverage. Progressive distillation also brings an additional benefit, *i.e.*, for multi-step sampling, our model predicts the next location on the ODE trajectory instead of jumping to the ODE endpoints every time by other distillation approaches [58, 66, 75]. This better preserves the original model behavior and facilitates better compatibility with LoRA modules [16] and control plugins [10, 74, 76].

Furthermore, our paper proposes innovative discriminator design, loss objectives, and stable training techniques. Specifically, we use the pre-trained diffusion UNet encoder as the discriminator backbone and fully operate in latent space. We propose two adversarial loss objectives to trade off sample quality and mode coverage. We investigate the implication of diffusion schedules and output formulation. We discuss techniques to stabilize the adversarial training.

Our distillation method produces new state-of-the-art SDXL models that support one-step/few-step generation at 1024px resolution. We open-source our distilled models as SDXL-Lightning.

2. Background

2.1. Diffusion Model

The forward diffusion process [14] gradually transforms samples from the data distribution to the Gaussian noise distribution. Given a data sample x_0, noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and time $t \sim \mathcal{U}(1, T)$. The forward function is defined as the following, with $\bar{\alpha}_t$ as the manually defined schedule [14]:

$$x_t = \mathbf{forward}(x_0, \epsilon, t) \equiv \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad (1)$$

A neural network $f : \mathbb{R}^d \to \mathbb{R}^d$ is trained to predict the gradient field u_t at any location x_t of the flow. The network is conditioned on time t and optionally other conditions c:

$$\hat{u}_t = f(x_t, t, c) \quad (2)$$

Many prior works formulate the network to perform noise prediction [14], *i.e.* $\hat{\epsilon} = f(x_t, t, c)$. We can use the conversion function $\mathbf{x}$ to convert the prediction to $\hat{x}_0$ space:

$$\hat{x}_0 = \mathbf{x}(x_t, \hat{\epsilon}, t) \equiv (x_t - \sqrt{1 - \bar{\alpha}_t} \hat{\epsilon}) / \sqrt{\bar{\alpha}_t} \quad (3)$$

Alternatively, the network can be formulated to perform data sample prediction [14], *i.e.* $\hat{x}_0 = f(x_t, t, c)$. We can use the conversion function ϵ to convert the prediction to $\hat{\epsilon}$ space:

$$\hat{\epsilon} = \boldsymbol{\epsilon}(x_t, \hat{x}_0, t) \equiv (x_t - \sqrt{\bar{\alpha}_t} \hat{x}_0) / \sqrt{1 - \bar{\alpha}_t} \quad (4)$$

Regardless of the formulation, the network in essence predicts the gradient $\hat{u}_t$. Given the gradient u_t at any location x_t, we can move samples along the flow:

$$\begin{aligned} x_{t'} &= \mathbf{move}(x_t, u_t, t, t') \\ &\equiv \mathbf{forward}(\mathbf{x}(x_t, u_t, t), \boldsymbol{\epsilon}(x_t, u_t, t), t') \end{aligned} \quad (5)$$

The generation process is simply moving sample $x_T \sim \mathcal{N}(0, \mathbf{I})$ from $t = T$ to $t = 0$ a small step at a time.

2.2. Latent Diffusion Model

Instead of directly generating samples at the data space, latent diffusion models (LDMs) [51] propose to first train a Variational Autoencoder (VAE) [25] that encodes the data to a more compact latent space. Diffusion models are trained to generate the latent codes, which are passed through the VAE decoder to generate the final data sample.

Latent diffusion models are widely adopted for high-resolution image and video generation due to their computational efficiency. SDXL [44] is the state-of-the-art text-to-image generation model that can generate 1024px resolution images from 128px latent space.

2.3. Progressive Distillation

Progressive distillation [54] trains the student to predict directions pointing to the next flow location as if the teacher has performed multiple steps.

Specifically, given data x_0, c from the dataset, and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, we jump to arbitrary timestep t:

$$x_t = \mathbf{forward}(x_0, \epsilon, t) \tag{6}$$

We use the frozen teacher network f_{teacher} to perform n inference steps to derive x_{t-ns} ($t - ns$ clamped to $[0, T]$):

$$u_t = f_{\text{teacher}}(x_t, t, c) \tag{7}$$
$$x_{t-s} = \mathbf{move}(x_t, u_t, t, t-s) \tag{8}$$
$$u_{t-s} = f_{\text{teacher}}(x_{t-s}, t-s, c) \tag{9}$$
$$x_{t-2s} = \mathbf{move}(x_{t-s}, u_{t-s}, t-s, t-2s) \tag{10}$$
$$\cdots \tag{11}$$
$$u_{t-(n-1)s} = f_{\text{teacher}}(x_{t-(n-1)s}, t-(n-1)s, c) \tag{12}$$
$$x_{t-ns} = \mathbf{move}(x_{t-(n-1)s}, u_{t-(n-1)s}, t-(n-1)s, t-ns) \tag{13}$$

Then, we train the student network f_{student} to predict a direction field $\hat{u}_t$ that points from x_t directly to x_{t-ns}:

$$\hat{u}_t = f_{\text{student}}(x_t, t, c) \tag{14}$$
$$\hat{x}_{t-ns} = \mathbf{move}(x_t, \hat{u}_t, t, t-ns) \tag{15}$$

The original work uses MSE loss [54]:

$$\mathcal{L}_{\text{mse}} = \|\hat{x}_{t-ns} - x_{t-ns}\|_2^2 \tag{16}$$

Once the student model converges, it is used as the teacher model and the distillation process repeats. In theory, it can produce one-step generation models, but in practice, models produce blurry results. We analyze this issue in Section 3.1.

2.4. Adversarial Distillation

Adversarial training involves a minimax optimization between a discriminator network that aims to identify generated samples from real samples and a generator network that aims to fool the discriminator. It was originally proposed as Generative Adversarial Networks (GANs) [8], a standalone class of generative networks, but it suffers from issues such as mode collapse and instability. Recent studies have found that the adversarial objective can be incorporated in diffusion training [72] and distillation [58, 73].

SDXL-Turbo [58] is the latest and the most popular open-source model using adversarial diffusion distillation. It follows prior works [56, 57] to use a pre-trained image encoder DINOv2 [41] as the discriminator backbone to accelerate training. However, this brings several limitations. First, using an off-the-shelf vision encoder means it must operate in the pixel space instead of the latent space, which significantly increases computation, memory consumption,

and training time, making high-resolution distillation impractical. This is likely the reason SDXL-Turbo only supports up to 512px resolution. Second, an off-the-shelf vision encoder only works at $t = 0$. The distilled model has to be trained to jump to ODE endpoints x_0, but since the quality for one-step inference is not good enough, random noises are added again for multi-step inference. This way of multi-step inference significantly alters the model behavior, making it less compatible with existing LoRA modules [16] and control plugins [10, 74, 76]. Third, off-the-shelf encoders may be hard to find for other datasets (anime, line arts, *etc.*) and modalities (video, audio, *etc.*). This reduces the generalizability of the distillation method. Lastly, the adversarial objective alone does not force the model to follow the same ODE flow, so mode coverage is not enforced.

Our method uses the diffusion model's U-Net encoder as the discriminator backbone. This allows us to efficiently distill in the latent space for high-resolution models, supports discrimination at all timesteps, and is generalizable to all datasets and modalities. Our method also allows control over the trade-off between quality and mode coverage, as later discussed in Sections 3.2 and 3.4.

2.5. Other Distillation Methods

We briefly discuss the advantages of our approach compared to other distillation methods.

Consistency Model (CM) [65, 66] also requires jumping to the ODE endpoints at every inference step. This causes large model behavior changes for multi-step sampling which reduces compatibility with LoRA modules and plugins. This method has been applied to SDXL [36, 37] but its generation quality is poor under 8 steps. Consistency Trajectory Model (CTM) [22] adds adversarial loss and supports jumping to arbitrary flow locations, but the adversarial training is applied post-distillation, instead of during the distillation, and the method has not been applied to large-scale text-to-image models.

Rectified Flow (RF) [31, 32] straightens the flow by repeatedly training with deterministic data and noise pairs. However, its few-step generation quality is still poor. Additionally, since the model has only seen specific data and noise pairs during the distillation, it no longer supports data pairing with arbitrary noise. This impacts the ability for image editing such as SDEdit [38].

Score Distillation Sampling (SDS) [45] has been used in SDXL-Turbo [58] to stabilize adversarial training, yet its effect is minimal and it cannot be used as a distillation method alone. Variation Score Distillation (VSD) [70] has recently been used in diffusion distillation [75]. However, it requires training an additional score model of the negative distribution during the distillation process, and like the discriminator in adversarial training, it also involves a dynamic training target that can negatively affect training stability. There

is no open-source model for comparison, and our preliminary experiments find our method achieves better quality.

2.6. LoRA

Low-Rank Adaptation (LoRA) [16] is an efficient fine-tuning technique. It only trains a small number of additional parameters to the model and has become particularly popular for training stylization modules for existing text-to-image models.

LCM-LoRA [37] is the first to show that model distillation can also be trained as a LoRA module. This ensures minimum parameter changes and can be conveniently plugged into the existing ecosystem.

Our work is inspired by this approach and we provide our distilled models both as LoRAs for convenient plug and play and as full models for even better quality.

3. Method
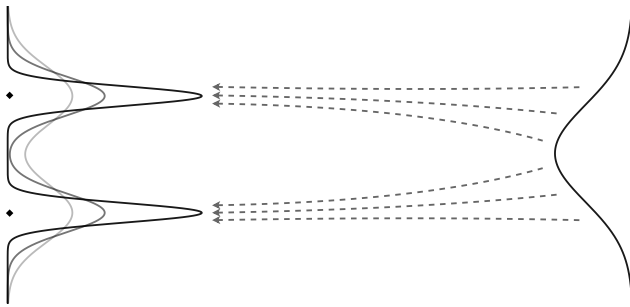
3.1. Why Distillation with MSE Fails

Figure 1. Illustration of multiple possible flows learned by models with different capacities. Distilled student models for few-step generations do not have the same capacity to match with the teacher models, leading to blurry results with MSE loss.

The learned ODE flow is determined by the forward schedule [29, 31], the loss function [27], and the model capacity. Given finite training samples, the underlying data distribution is ambiguous. The maximum likelihood estimation (MLE) is a distribution that assigns even probability only to the observed samples and zero everywhere else. If the model has infinite capacity, it will learn a flow of this maximum likelihood estimation and overfit to always produce observed samples and generate no new data. In practice, diffusion models can generate new data because neural networks are not perfect learners.

When the model is used in multi-step generations, it is stacked and has a higher Lipschitz constant and more non-linearities to approximate a more complex distribution. But when the model is used in few-step generations, it no longer has the same amount of capacity to approximate well the same distribution. This is evidenced by diffusion models can have very sharp changes in results despite small changes

in the initial noises [9], but the distilled models have much smoother latent traversal.

This explains why distillation with MSE loss produces blurry results. The student model simply does not have the capacity to match the teacher. We find that other distance metrics, such as L1 and perceptual loss [27, 77], also produce undesirable results. On the other hand, we find adversarial objectives to be effective in mitigating this issue.

3.2. Adversarial Objective

Instead of using the MSE loss between the student-predicted $\hat{x}_{t-ns}$ and teacher-predicted x_{t-ns} as in Equation (16), we use an adversarial discriminator. Specifically, our discriminator $D : \mathbb{R}^d \to \mathbb{R} \in [0, 1]$ computes the probability of x_{t-ns} being generated from the teacher as opposed to the student, given condition x_t and c.

$$D(x_t, x_{t-ns}, t, t-ns, c) \qquad (17)$$

We use non-saturated adversarial loss [8] and train the discriminator and the student model in alternating steps. This encourages the student prediction $\hat{x}_{t-ns}$ to be closer to the teacher prediction x_{t-ns}:

$$p = D(x_t, x_{t-ns}, t, t-ns, c) \qquad (18)$$
$$\hat{p} = D(x_t, \hat{x}_{t-ns}, t, t-ns, c) \qquad (19)$$
$$\mathcal{L}_D = -\log(p) - \log(1-\hat{p}) \qquad (20)$$
$$\mathcal{L}_G = -\log(\hat{p}) \qquad (21)$$

The condition on x_t is important for preserving the ODE flow. This is because the teacher's generation of x_{t-ns} is deterministic from x_t. By providing the discriminator both x_{t-ns} and x_t, the discriminator learns the underlying ODE flow and the student must also follow the same flow to fool the discriminator.

Our formulation is very similar to a prior work [72] except we use it for distillation instead of training from scratch. Note that this approach only preserves the ODE flow and ensures mode coverage when used in distillation.

3.3. Discriminator Design

A prior work [27] has shown that a pre-trained diffusion model's U-Net [52] encoder can be used as a vision backbone. Such a pre-trained backbone is very suitable for our discriminator because it has been pre-trained on the target dataset, directly operates in the latent space, supports noised input at all timesteps, and supports text condition.

We follow the approach and copy the encoder and midblock of the pre-trained SDXL model as our discriminator backbone d. We pass x_{t-ns} and x_t independently through the shared backbone d, concatenate the hidden features after the midblock in the channel dimension, and pass it to a prediction head. The prediction head consists of simple

blocks of 4×4 convolution with a stride of 2, group normalization [71] with 32 groups, and SiLU activation [11,48] layers to further reduce the spatial dimension. The output is projected to a single value and clamped to $[0, 1]$ range with sigmoid $\sigma(\cdot)$. Together they form the complete discriminator D:

$$
\begin{aligned}
&D(x_t, x_{t-ns}, t, t - ns, c) \\
&\equiv \sigma\Big(\text{head}\big(d(x_{t-ns}, t - ns, c), d(x_t, t, c) \big) \Big)
\end{aligned} \quad (22)
$$

Note that the backbone is initialized with the pre-trained weights and we train the entire discriminator without freezing the backbone. We find our training stable without the need for expansive R1 regularization [39] nor switching to L2 attention [17, 23]. Additional stabilization techniques are discussed in Section 3.7.

3.4. Relax the Mode Coverage

Figure 2. "Janus" artifacts appear when the student network does not have the capacity to match the teacher's sudden changes. This problem can be mitigated by relaxing the mode coverage requirement.

The adversarial objective above encourages the prediction to be both sharp and flow-preserving, but this does not change the fact that the student does not have enough capacity to perfectly match the teacher as discussed in Section 3.1. With the MSE objective, it manifests blurry results. With the adversarial objective, it manifests the "Janus" artifacts.

As shown in Figure 2, the teacher model can sometimes generate drastic layout changes for adjacent noise inputs, but the student model does not have the same capacity to make such sharp changes. As a result, the adversarial loss sacrifices semantic correctness in need to preserve the sharpness and the layout, manifesting artifacts that feature conjoined heads and bodies.

Semantic correctness is more important than mode coverage by human preference. Therefore, after training with the original adversarial objective, we relax the flow preservation requirement. Specifically, we further finetune the model without the condition on x_t:

$$
\begin{aligned}
&D'(x_{t-ns}, t - ns, c) \\
&\equiv \sigma\Big(\text{head}\big(d(x_{t-ns}, t - ns, c) \big) \Big)
\end{aligned} \quad (23)
$$

We find that finetuning with this objective is effective in removing the "Janus" artifacts while still preserving the original flow to a great extent in practice. Therefore, at every stage of the progressive distillation, we first train with the conditional objective and then finetune with this unconditional objective. Since the unconditional objective only concerns per-sample quality, we use the skip-level teacher for distilling the one-step and two-step models to further retain quality and mitigate error accumulation.

3.5. Fix the Schedule

A prior work [26] has shown that common diffusion schedules are flawed. Specifically, the schedule does not reach pure noise at $t = T$ during training, yet pure noise is given during inference, causing a discrepancy. Unfortunately, SDXL uses this flawed schedule. The effect is less obvious under a large number of inference steps but is particularly detrimental for few-step generations.

A hacky way to circumvent the problem is to hard swap pure noise ϵ as model input at $t = T$ during training. This way the model is trained to expect pure noise as input at $t = T$ and we still use Equation (3) with the old $\bar{\alpha}$ schedule at inference to avoid singularity. It incurs minimum changes to the sampling procedure with existing software ecosystems [69]. This approach is also used by SDXL-Turbo [58].

$$
\begin{aligned}
&\textbf{Forward}(x_0, \epsilon, t) \\
&= \begin{cases} \textbf{forward}(x_0, \epsilon, t), & \text{when } t < T \\ \epsilon, & \text{when } t = T \end{cases}
\end{aligned} \quad (24)
$$

3.6. Distillation Procedure

First, we perform distillation from 128 steps directly to 32 steps with MSE loss. We find MSE is sufficient for the early stage. We also apply classifier-free guidance (CFG) [15] only in this stage. We use a guidance scale of 6 without any negative prompts.

Then, we switch to using adversarial loss to distill the step count in this order: $32 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$. At each stage, we first train with the conditional objective as in Section 3.2 to preserve the ODE flow, and then train with the unconditional objective as in Section 3.4 to relax the mode coverage.

At each stage, we first train with LoRA using the two objectives, then we merge the LoRA and train the whole UNet further with the unconditional objective. We find finetuning the whole UNet can achieve even better performance, while the LoRA module can be used on other base models. Our LoRA settings are the same as LCM-LoRA [37], which uses rank 64 on all the convolution and linear weights except the input and output convolutions and the shared time embedding linear layers. We do not use LoRA on the discriminator. We re-initialize the discriminator at each stage.

We distill our models on a subset of LAION [59] and COYO [4] dataset. We select images to be greater than 1024px and LAION images with aesthetic scores above 5.5. We additionally filter images by sharpness using a Laplacian filter and clean up the text prompts. The distillation is conducted on a square aspect ratio, but we find it generalizes well to other aspect ratios at inference time.

We use batch size 512 across 64 A100 80G GPUs. For the first $128 \rightarrow 32$ stage with MSE loss, we use learning rate 1e-5 with Adam $\beta_1 = 0.9, \beta_2 = 0.999$. For the remaining stages with adversarial loss, we use learning rates 1e-6 with LoRA and 5e-7 without LoRA for both the student and the discriminator networks. The Adam optimizer [24] uses $\beta_1 = 0, \beta_2 = 0.99$ following prior works [17, 21] without weight decay [33]. We use gradient accumulation, VAE slicing, BF16 mixed precision [40], flash attention [6, 7], and zero redundancy optimizer [47] to reduce the memory footprint.

3.7. Stable Training Techniques

For one-step and two-step distillations, we employ additional techniques to stabilize the training.

3.7.1 Train Student Networks at Multiple Timesteps

While we only need to train the one-step model at timestep $\{1000\}$, and the two-step model at timesteps $\{500, 1000\}$ for complete image generation, we find training on more timesteps $\{250, 500, 750, 1000\}$ improves stability. As an additional benefit, this allows our models to support SDEdit [38] at different timesteps as illustrated below:

One-Step Model   Two-Step Model
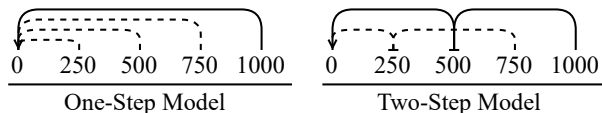
3.7.2 Train Discriminator at Multiple Timesteps

We find training the one-step model using the above discriminator formulation very unstable. We find the root reason is that our discriminator uses the pre-trained diffusion UNet encoder as the backbone, yet the diffusion encoder is trained to only focus on high-frequency details at lower timesteps and low-frequency structures at higher timesteps. For one-step generations, the student network directly predicts $\hat{x}_0$. If we pass $\hat{x}_0$ and $t = 0$ to the discriminator backbone, it is not able to critique the image structure. This leads to images with bad shapes and even divergence.

Our solution is to add noise to both teacher-predicted x_0 and student-predicted $\hat{x}_0$ to timesteps: $\{10, 250, 500, 750\}$ randomly. This way the discriminator can critique the prediction on both high-frequency details and low-frequency structures.

Specifically, we first draw $t* \leftarrow \{10, 250, 500, 750\}$ with uniform weighting 1:1:1:1 and sample a new noise $\epsilon* \sim \mathcal{N}(0, \mathbf{I})$. Then we apply the noise before passing it through the discriminator:

$$D(x_t, \mathbf{forward}(\hat{x}_0, \epsilon*, t*), t, t*, c) \tag{25}$$

$$D'(\mathbf{forward}(\hat{x}_0, \epsilon*, t*), t*, c) \tag{26}$$

After the model is trained stable, we change the timesteps weighting to 5:1:1:1. This further improves details and removes noisy artifacts.

Note that this stabilization technique can also be viewed from the lens of bridging the distribution gap [39], discriminator augmentation [20], and multi-scale discriminator [18].

3.7.3 Switch to x_0 Prediction

We find the one-step model with ϵ-prediction formulation tends to generate noise artifacts likely due to numerical instability. We change the one-step model to x_0-prediction and it resolves the issue.

Specifically, we copy the network and convert the predicted $\hat{\epsilon}$ to $\hat{x}_0$ through conversion function $\mathbf{x}$ defined in Equation (3). We use MSE to gradually guide the online model to x_0-prediction.

$$\hat{\epsilon} = f_{\text{frozen}}(x_t, t, c) \tag{27}$$

$$\hat{x}_0 = f_{\text{online}}(x_t, t, c) \tag{28}$$

$$\mathcal{L}_{\text{convert}} = \|\hat{x}_0 - \mathbf{x}(x_t, \hat{\epsilon}, t)\|_2^2 \tag{29}$$

After the conversion, the one-step model is trained with adversarial objectives in x_0-prediction formulation, while the teacher model still operates in ϵ-prediction formulation. Due to the substantial formulation change, we do not provide LoRA for one-step generation.

4. Evaluation

4.1. Specification Comparison

Table 1 shows the specification of our distilled models compared to others.

Method	Steps Needed	Resolution	CFG Free	Offer LoRA
SDXL [44]	25+	1024px	No	-
LCM [36, 37]	4+	1024px	Yes&No	Yes
Turbo [58]	1+	512px	Yes	No
Ours	**1+**	**1024px**	**Yes**	**Yes**

Table 1. Model specifications. Our method requires the fewest amount of steps to produce high-quality samples.

SDXL [44]	Ours				Turbo [58]		LCM [36]	
64NFE, CFG6	No CFG				512px, No CFG		8NFE, CFG8	No CFG
32 Steps	8 Steps	4 Steps	2 Steps	1 Step	4 Steps	1 Step	4 Steps	1 Step

(a) An Asian firefighter with a rugged jawline rushes through the billowing smoke of an autumn blaze.

(b) A close-up of an Asian lady with sunglasses.

(c) The 90s, a beautiful woman with a radiant smile and long hair, dressed in summer attire.

(d) A majestic lion stands proudly on a rock, overlooking the vast African savannah.

(e) A monkey making latte art.

(f) In a fantastical scene, a creature with a human head and deer body emanates a green light.

(g) A delicate porcelain teacup sits on a saucer, its surface adorned with intricate blue patterns.

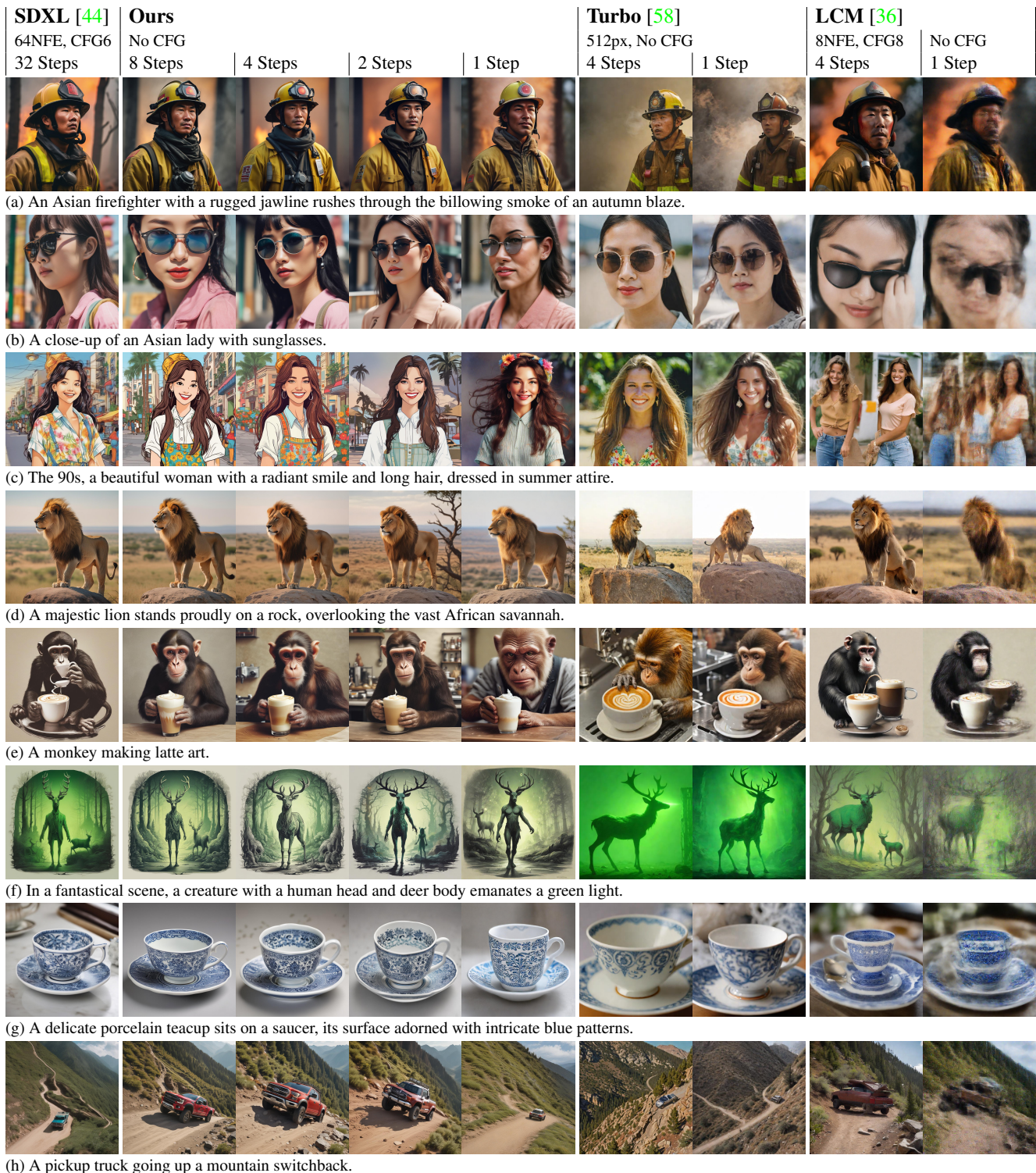(h) A pickup truck going up a mountain switchback.

Figure 3. Qualitative comparison. Our models here are fully trained instead of LoRA. Models are given the same initial noise for each prompt, except SDXL-Turbo since it only supports 512px resolution. Our model produces the best results in all prompts, and it best preserves the style and layout of the original model. Note 1: Original SDXL and ours use Euler sampler while other methods use their default samplers. Note 2: some methods require classifier-free guidance (CFG) at inference, which doubles the number of function evaluations (NFE) and doubles the computation. (Please zoom in to view at the full resolution.)
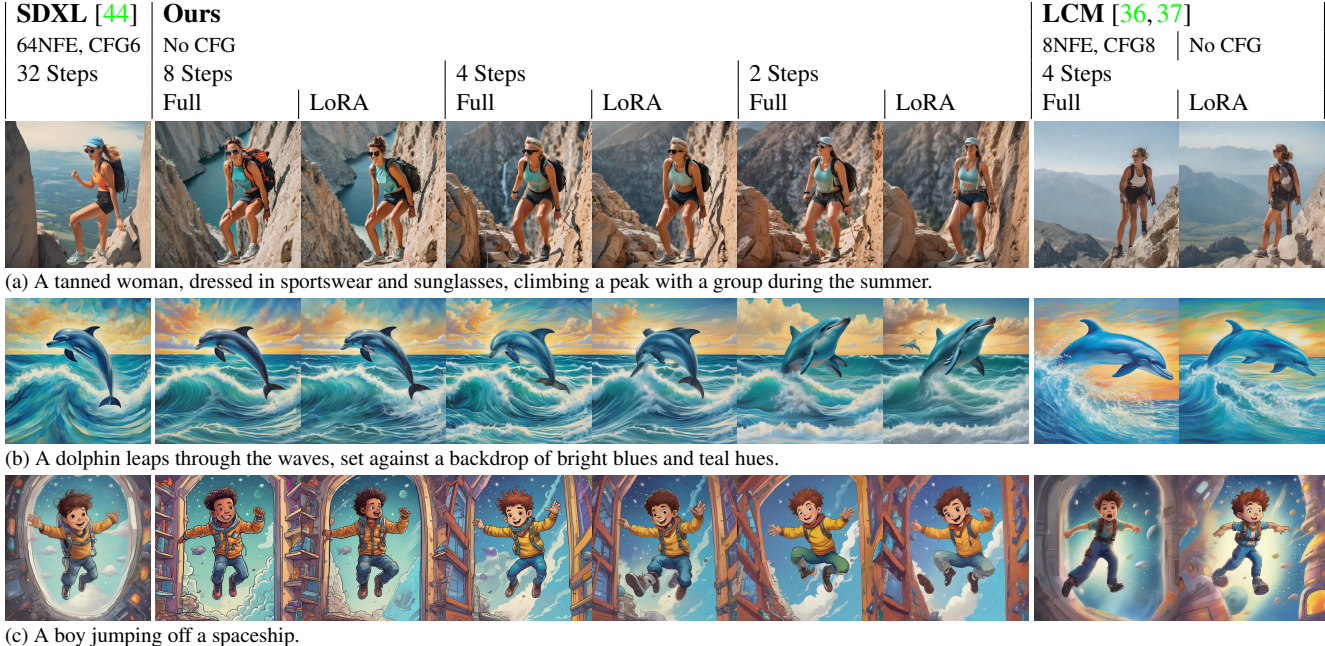
SDXL [44]	Ours						LCM [36, 37]	
64NFE, CFG6	No CFG						8NFE, CFG8	No CFG
32 Steps	8 Steps		4 Steps		2 Steps		4 Steps	
	Full	LoRA	Full	LoRA	Full	LoRA	Full	LoRA

(a) A tanned woman, dressed in sportswear and sunglasses, climbing a peak with a group during the summer.

(b) A dolphin leaps through the waves, set against a backdrop of bright blues and teal hues.

(c) A boy jumping off a spaceship.

Figure 4. Comparison between fully-trained and LoRA-trained models. Training the full UNet has slightly better details, but our LoRA models are also very high quality. Note we do not provide 1-step LoRA. (Please zoom in to view at the full resolution.)

4.2. Qualitative Comparison

Figure 3 compares our method against other open-source distillation models: SDXL-Turbo [58] and LCM [36]. Our method is substantially better in overall quality and details. Our method is also substantially better in the preservation of the style and layout of the original model. Furthermore, we find our 4-step and 8-step model can often outperform the original SDXL model for 32 steps. This is because our progressive distillation starts all the way from 128 steps.

Figure 4 compares our LoRA models against the fully trained models. We find that fully trained models have better structures and details. This is less noticeable on 8-step models, but more observable on 2-step models.

4.3. Quantitative Comparison

Table 2 shows Fréchet Inception Distance (FID) [12, 42] and CLIP score [46]. Following the convention, we generate images using the first 10K prompts from the COCO [28] validation dataset. FID metric is computed against the corresponding ground truth images from COCO.

FID is normally computed by resizing the whole image to 299px for the InceptionV3 network [68]. This only assesses the high-level sample quality and diversity. The metric shows that our model achieves similar performance as other distillation techniques. All distillation methods have worse FID compared to the original SDXL likely due to the reduction in diversity.

We additionally propose to calculate FID on patches of images to assess high-resolution details. Specifically, we

Method	Steps	FID ↓ (Whole)	FID ↓ (Patch)	CLIP ↑
SDXL [44]	32	18.49	35.89	26.48
LCM [36]	1	80.01	158.90	23.65
LCM [36]	4	21.85	42.53	26.09
LCM-LoRA [37]	4	21.50	40.38	26.18
Turbo [58]	1	23.71	43.69	26.36
	4	22.58	42.65	26.18
Ours	1	22.61	**41.53**	26.02
	2	23.11	**35.12**	25.98
	4	22.30	**33.52**	26.07
	8	21.43	**33.55**	25.86
Ours-LoRA	2	23.39	**40.54**	26.18
	4	23.01	**34.10**	26.04
	8	22.30	**33.92**	25.77

Table 2. Quantitative comparison. FID-Whole reflects high-level diversity and quality. FID-Patch reflects high-resolution details. CLIP score reflects text-alignment. Our models have significantly better high-resolution details while retaining similar performance in diversity and text alignment.

calculate FID on the 299px center-cropped patch of every image. For Turbo, we resize the 512px to 1024px before the crop for a fair comparison. The metric shows that our models have much better high-resolution details compared to other methods. Additionally, the metric shows that our model has better high-resolution details compared to origi-

nal SDXL models for 32 steps because our distillation starts from 128 steps. It also shows that the quality degrades as the number of inference steps decreases.

CLIP score shows that our method achieves similar text-alignment performance compared to other methods.

5. Ablation

5.1. Apply LoRA on Other Base Models

Figure 5 shows that our distillation LoRA model can be applied to different base models. Specifically, we test it on third-party cartoon [55], anime [1], and realistic [50] base models. Our distillation LoRAs are able to keep the style and layout of the new base model to a great extent.
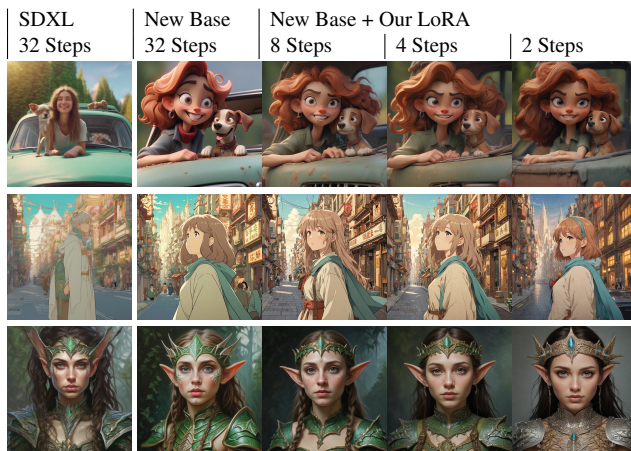
Figure 5. Our distillation LoRA can be applied to other base models, *e.g.* cartoon [55], anime [1], and realistic [50] base models.

5.2. Inference with Different Aspect Ratios

Figure 6 shows that are models can mostly retain the ability to infer at different resolutions and aspect ratios despite the distillation is only performed on square images. However, we do notice an increasing amount of bad cases when performing 1-step and 2-step generations. This can be improved by distilling with multiple aspect ratios, which we leave for future improvements.

Figure 6. Our model is trained only on square images but still can generate different aspect ratios. The example images are 1:2 aspect ratio, 720×1440px, generated by our 4-step model.

5.3. Compatibility with ControlNet

Figure 7 shows that our models are compatible with ControlNet [76]. We test it on the canny edge [60] and depth [61] ControlNet. We observe that our models follow the condition correctly, with some quality degradation as the number of inference steps decreases.
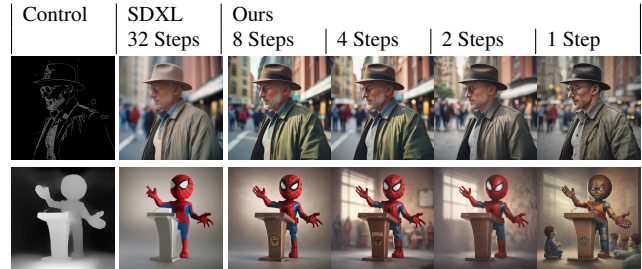

Figure 7. Our models are compatible with ControlNet [76]. Examples shown are generation conditioned on canny edge and depth.

6. Limitation

Unlike other methods [36,37,58] having a single distilled checkpoint that supports multiple inference step settings, our method produces separate checkpoints for each corresponding inference step setting. This is usually not an issue in production when the number of inference steps is fixed. In case the number of inference steps must be flexible, our LoRA modules can mitigate the checkpoint switching issue.

Our method produces distilled student models with the same architecture as the teacher model. However, we believe that the UNet architecture is not optimal for one-step generation. We inspect the feature maps at each UNet layer and find that most of the generation is carried out by the decoder. We leave this problem to future improvements.

7. Conclusion

To sum up, we have presented SDXL-Lightning, our state-of-the-art one-step/few-step text-to-image generative models resulting from our novel progressive adversarial diffusion distillation method. In our evaluation, we have found that our models produce superior image quality compared to prior works. We are open-sourcing our models to advance the research in generative AI.

References

[1] AAM-XL Anime Mix. https://civitai.com/models/269232. 9

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. 1

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023. 1

[4] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 6

[5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 1

[6] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. 6

[7] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. 6

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 3, 4

[9] Jiayi Guo, Xingqian Xu, Yifan Pu, Zanlin Ni, Chaofei Wang, Manushree Vasu, Shiji Song, Gao Huang, and Humphrey Shi. Smooth diffusion: Crafting smooth latent spaces in diffusion models, 2023. 4

[10] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023. 1, 2, 3

[11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 5

[12] 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. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. 8

[13] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022. 1

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 5

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2, 3, 4

[17] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis, 2023. 5, 6

[18] Animesh Karnewar and Oliver Wang. MSG-GAN: multiscale gradients for generative adversarial networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7796–7805. IEEE, 2020. 6

[19] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. 2

[20] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 6

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. IEEE, 2020. 6

[22] 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, 2023. 2, 3

[23] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 2021. 5

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[25] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2

[26] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed, 2023. 5

[27] Shanchuan Lin and Xiao Yang. Diffusion model with perceptual loss, 2024. 4

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 8

[29] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 2, 4

[30] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2

[31] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 2, 3, 4

[32] 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, 2023. 2, 3

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 6

[34] 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, 2022. 2

[35] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. 2

[36] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023. 2, 3, 6, 7, 8, 9

[37] 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, 2023. 2, 3, 4, 5, 6, 8, 9

[38] 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3, 6

[39] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 2018. 5, 6

[40] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 6

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 3

[42] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation, 2022. 8

[43] Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Wuerstchen: An efficient architecture for large-scale text-to-image diffusion models, 2023. 1

[44] 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, 2023. 1, 2, 6, 7, 8

[45] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 3

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 8

[47] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models, 2020. 6

[48] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. 5

[49] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[50] RealVisXL V4.0. https://civitai.com/models/139562. 9

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1

[54] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2, 3

[55] Samaritan 3D Cartoon V4. https://civitai.com/models/81270. 9

[56] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17480–17492, 2021. 3

[57] 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, 2023. 3

[58] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. 2, 3, 5, 6, 7, 8, 9

[59] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 6

[60] SDXL-ControlNet Canny. https://huggingface.co/diffusers/controlnet-canny-sdxl-1.0. 9

[61] SDXL-ControlNet Depth. https://huggingface.co/diffusers/controlnet-depth-sdxl-1.0. 9

[62] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 1

[63] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015. 1

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[65] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models, 2023. 2, 3

[66] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. 2, 3

[67] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1, 2

[68] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. 8

[69] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 5

[70] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation, 2023. 3

[71] Yuxin Wu and Kaiming He. Group normalization, 2018. 5

[72] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3, 4

[73] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans, 2023. 2, 3

[74] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ipadapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. 2, 3

[75] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation, 2023. 2, 3

[76] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 3, 9

[77] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. IEEE Computer Society, 2018. 4

[78] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models, 2023. 2

[79] Chuanxia Zheng, Long Tung Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation, 2022. 1

[80] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models, 2023. 1