Single-Step Diffusion via Direct Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce Direct Models, a generative modeling framework that enables single-step diffusion by learning a direct mapping from initial noise x_0 to all intermediate latent states along the generative trajectory. Unlike traditional diffusion models that rely on iterative denoising or integration, Direct Models leverages a progressive learning scheme where the mapping from x_0 to $x_{t+\delta t}$ is composed as an update from x_0 to x_t plus the velocity at time t. This formulation allows the model to learn the entire trajectory in a recursive, data-consistent manner while maintaining computational efficiency. At inference, the full generative path can be obtained in a single forward pass. Experimentally, we show that Direct Models achieves state-of-the-art sample quality among single-step diffusion methods while significantly reducing inference time.

1 Introduction

Diffusion models and flow matching methods have recently achieved remarkable success across a wide range of applications, including image synthesis [5], audio generation [7], and 3D shape modeling [10]. Despite their effectiveness, a significant limitation of these approaches lies in their reliance on iterative sampling or inference procedures, which are computationally expensive and can limit real-time deployment.

To address this bottleneck, some recent works have explored distillation techniques to compress multi-step diffusion or flow matching models into efficient single-step samplers. Notable examples include [12, 16], which require first training a high-quality teacher model and then performing a costly distillation step. Moreover, such distillation sometimes involves constructing large synthetic datasets, increasing the complexity and resource demands of the overall pipeline.

In contrast, this work proposes a novel *single-run training* approach that directly learns a *one-step* diffusion-like generative model without relying on teacher models or distillation. Our method offers a more practical and efficient solution for fast sampling, avoiding the overhead inherent in multi-stage training pipelines and results in high-quality samples (see Figure 1).

To address these challenges, we propose a new class of *Direct Models*, a residual-based formulation that enables both single-step sampling and single-run training. The key idea is to directly model the full flow map through a time-indexed residual field, allowing us to query any intermediate latent x_t without requiring numerical integration. This direct access to latents motivates the name of our approach.

At the core of our method lies a simple recursive structure: the residual displacement at time $t + \delta t$ is expressed as a combination of the residual at time t and the local velocity at that point. This recursive formulation not only serves as a training objective but also acts as a structural prior, encouraging consistency across time steps while remaining efficient and fully self-supervised.

Our main contributions are as follows:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Figure 1: Generations of multi-step flow-matching models and single-step Direct Models. Top row: 128-step generation by a vanilla flow matching model. Bottom row: Generations with our single-step model. Direct Models generates high-quality images across a wide range of inference budgets, including using a single forward pass, drastically reducing sampling time by up to $128\times$ compared to diffusion and flow-matching models. The same starting noise is used within each column.

- We propose Direct Models, a novel direct residual model for one-step flow generation that enables efficient single-step sampling without iterative inference.
- We introduce a recursive training framework, based on a straightforward mathematical derivation based on a Taylor expansion of the flow, that enforces local velocity consistency, allowing the model to be trained in a single run.
- We demonstrate that Direct Models achieves superior sample quality compared to existing single-step, single-run generative approaches, closing the gap with iterative methods while maintaining fast inference.

2 Preliminaries: Continuous-Time Generative Models

Modern generative modeling has been significantly shaped by methods that transform simple source distributions into complex data distributions through continuous-time dynamics. Two prominent families in this space are *diffusion-based models* (*e.g.*, [18, 5, 19]) and *flow-based approaches*, particularly those based on *flow matching* [8, 9]. These frameworks parameterize sample trajectories using neural differential equations, typically in the form of an ODE, to transport mass smoothly from a source distribution (*e.g.*, Gaussian noise) to a target data distribution.

In this work, we adopt a *flow matching* viewpoint, leveraging its optimal transport-inspired formulation to model deterministic sample paths. Notably, recent studies (*e.g.*, [6]) have emphasized the close relationship between diffusion and flow-based models, observing that flow matching can be viewed as a deterministic instance of more general stochastic diffusion processes. As such, we view these paradigms as conceptually intertwined and refer to them in parallel where appropriate.

Formally, consider a pair of distributions: a base distribution μ_0 and a target distribution μ_1. The goal is to learn a *velocity field* $\mathbf{v}_\theta(x,t)$, parameterized by a neural network, that defines the evolution of a sample over time

$$\frac{d}{dt}\phi(x,t) = \mathbf{v}_\theta(\phi(x,t),t), \text{with } \phi(x,0) = x_0, \quad x_0 \sim \mu_0. \tag{1}$$

Solving this ODE from $t = 0$ to $t = 1$ generates a trajectory that ideally maps μ_0 into μ_1.

A practical and efficient instantiation of this idea is given by *Conditional Flow Matching (CFM)*, which sidesteps density estimation by using known correspondence pairs $(x_0, x_1) \sim (\mu_0, \mu_1)$. Rather than relying on stochastic score-based gradients, the model is trained to approximate the *ground-truth transport velocity* along straight-line paths

$$x_t = (1-t)x_0 + tx_1. \tag{2}$$

The true instantaneous velocity along this path is simply $x_1 - x_0$, and the model $\mathbf{v}_\theta(x,t)$ is trained to match this velocity using the loss

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,x_1,t}\left[\left\|\mathbf{v}_\theta\big((1-t)x_0 + tx_1, t\big) - (x_1 - x_0)\right\|^2\right]. \tag{3}$$

This supervised objective encourages the model to replicate the optimal displacement between samples at intermediate points in time, by constructing a continuous flow without requiring likelihoods or sampling noise. Once trained, generation consists of drawing $x_0 \sim \mu_0$ and integrating the ODE (1) forward using the learned dynamics. This process can be efficiently implemented with standard ODE solvers such as Euler or Runge–Kutta methods.

3 Method: One-Step Flow via Direct Models

3.1 Formulation

We introduce a one-step generative model by directly parameterizing the flow map $\phi(x,t)$. A natural formulation is to define the flow as $\phi(x,t) = x + w(x,t)$, where the residual field $w(x,t)$ captures the displacement from the initial point. However, this choice allows the magnitude of the displacement to vary arbitrarily with time which we found leading to unstable training. To impose a form of temporal consistency, we instead define the flow as

$$\phi(x,t) = x + t \cdot w(x,t), \tag{4}$$

where $w(x,t) \in \mathbb{R}^d$ is now interpreted as a *normalized direction of displacement*, and the scaling by t ensures that the overall displacement grows smoothly from zero to its final value. This parameterization encourages the magnitude $\|t \cdot w(x,t)\|$ to vary linearly with time, providing a stable and interpretable structure for learning. This formulation provides a single-step trajectory, in contrast to the continuous ODE integration approach commonly used in flow matching.

In the flow matching framework, the temporal derivative of the trajectory satisfies

$$\frac{d}{dt}\phi(x,t) = v(\phi(x,t),t), \tag{5}$$

where $v(x_t,t)$ is the target velocity field at the point $x_t = \phi(x,t)$. To incorporate this into our model, we compute the time derivative of $\phi(x,t)$ as defined in Equation (4)

$$\frac{d}{dt}\phi(x,t) = w(x,t) + t \cdot \frac{\partial w(x,t)}{\partial t}. \tag{6}$$

We approximate the time derivative of $w(x,t)$ using the forward difference with a discrete δt step

$$\frac{\partial w(x,t)}{\partial t} \approx \frac{w(x,t+\delta t) - w(x,t)}{\delta t}. \tag{7}$$

By substituting this approximation into the derivative of $\phi(x,t)$, we obtain

$$\frac{d}{dt}\phi(x,t) = w(x,t) + t \cdot \frac{w(x,t+\delta t) - w(x,t)}{\delta t}. \tag{8}$$

By matching this expression to the target velocity $v(x_t,t)$, we then have

$$w(x,t) + t \cdot \frac{w(x,t+\delta t) - w(x,t)}{\delta t} = v(x_t,t), \tag{9}$$

which can be rearranged into the equation

$$t \cdot \frac{w(x,t+\delta t) - w(x,t)}{\delta t} = v(x_t,t) - w(x,t). \tag{10}$$

Finally, by multiplying both sides by $\frac{\delta t}{t}$ we have

$$w(x,t+\delta t) - w(x,t) = \frac{\delta t}{t} \cdot (v(x_t,t) - w(x,t)), \tag{11}$$

and by isolating $w(x,t+\delta t)$, we arrive at

$$w(x,t+\delta t) = \frac{t-\delta t}{t} \cdot w(x,t) + \frac{\delta t}{t} \cdot v(x_t,t). \tag{12}$$

3.2 Training Direct Models via Local Velocity Propagation

To learn the flow map parameterized by a model w_ν, with parameters ν, we leverage the recursive structure implied by the progressive velocity propagation equation

$$\boxed{w(x, t + \delta t) = \frac{t - \delta t}{t} \cdot w(x, t) + \frac{\delta t}{t} \cdot v(x_t, t)}. \tag{13}$$

This relation connects the residual field w at two consecutive time steps through the velocity field v. We exploit this property to define a consistency-based training loss for w_ν, encouraging it to align with the propagated velocity information.

In our formulation, we train two models jointly:

- $v_\theta(x, t)$: a velocity field trained using the standard Conditional Flow Matching (CFM) loss,
- $w_\nu(x, t)$: a residual displacement field trained using a recursive propagation loss derived from Equation (13).

The velocity field v_θ is trained using the standard Conditional Flow Matching (CFM) loss. Given a sample pair $(x_0, x_1) \sim (\mu_0, \mu_1)$ and a uniformly sampled time $t \sim \mathcal{U}[0, 1]$, we define the intermediate point

$$x_t = (1 - t) \cdot x_0 + t \cdot x_1. \tag{14}$$

The CFM objective encourages the predicted velocity to match the ground-truth displacement between x_0 and x_1 at this intermediate point

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{x_0, x_1, t} \left[\| v_\theta(x_t, t) - (x_1 - x_0) \|^2 \right]. \tag{15}$$

The residual field w_ν is trained using a local velocity propagation loss derived from Equation (13). Given the sample $x_0 \sim \mu_0$, a small step size δt and $t' \sim \mathcal{U}[\delta t, 1 - \delta t]$, we define the propagation loss as

$$\mathcal{L}_{\text{prop}}(\nu) = \mathbb{E}_{x_0, t'} \left[\left\| w_\nu(x_0, t' + \delta t) - \left(\frac{t' - \delta t}{t'} \cdot \text{sg}[w_\nu(x_0, t')] + \frac{\delta t}{t'} \cdot v_\theta(x'_t, t') \right) \right\|^2 \right], \tag{16}$$

with $x'_t = x_0 + t' \cdot \text{sg}[w_\nu(x_0, t')]$, where $\text{sg}[\cdot]$ denotes a stop-gradient operator. Notice that we define the residual field model w_ν only with respect to the samples x_0 from the initial distribution μ_0. In this way, at inference we can directly map these samples to any point along the trajectory in one step, and, in particular, to the target distribution samples x_1. Although $w_\nu(x_0, t')$ may be initially uninformative at the beginning of the training, the propagation loss remains effective, removing the need for explicit scheduling of t'. This simplifies training, improving stability and practicality without compromising performance.

Our training algorithm is outlined in Algorithm 1.

3.3 Sampling from Direct Models

Sampling from our direct flow map model is straightforward and efficient. Given an initial sample $x_0 \sim \mu_0$, the corresponding transformed sample x_1 can be obtained via a single forward pass of the residual field

$$x_1 = x_0 + w_\nu(x_0, 1). \tag{17}$$

This one-step sampling eliminates the need for iterative procedures, making the approach practical and fast for inference.

4 Experiments

4.1 Settings

In this section, we compare our method against several existing approaches. All models are trained from scratch using the same architecture and implementation framework to ensure a fair comparison. Specifically, we adopt the DiT-B diffusion transformer architecture from [14]. Our experiments

Algorithm 1 Training Direct Models via Local Velocity Propagation

Initialize parameters θ for v_θ, ν for w_ν
for each training step **do**
    Sample pair $(x_0, x_1) \sim (\mu_0, \mu_1)$
    Train velocity model v_θ with CFM loss:
    Sample $t \sim \mathcal{U}[0, 1]$
    Compute $x_t = (1-t)x_0 + tx_1$ and
    Minimize

$$\mathcal{L}_{\text{CFM}} = \|v_\theta(x_t, t) - (x_1 - x_0)\|^2$$

    with respect to θ
    Update θ
    Train residual field w_ν with local propagation loss:
    Sample $t' \sim \mathcal{U}[\delta t, 1 - \delta t]$
    Compute $x'_t = x_0 + t' \cdot \text{sg}[w_\nu(x_0, t')]$
    Minimize

$$\mathcal{L}_{\text{prop}} = \left\| w_\nu(x_0, t' + \delta t) - \left(\frac{t' - \delta t}{t'} \cdot \text{sg}[w_\nu(x_0, t')] + \frac{\delta t}{t'} \cdot v_\theta(x'_t, t') \right) \right\|^2$$

    with respect to ν
    Update ν
end for

include unconditional generation on the CelebAHQ-256 dataset [11] and we also provide a comparison with class-conditional generation on ImageNet-256 [2] . For the results in Table 1 , we use the AdamW optimizer with a fixed learning rate of 5×10^{-5} and no weight decay. Models are trained for 500K iterations, using a step size of $\delta t = 10^{-2}$ and batch size of 64. Additionally, all models operate in the latent space provided by the `sd-vae-ft-mse` autoencoder [15].

4.2 Compared Methods

We compare our method to several prior approaches, following the same comparison setup as in [3]. For completeness, we briefly describe the training details of the compared methods based on the descriptions in [3].

We consider two categories of diffusion-based models: distillation methods,[1] which involve pretraining a diffusion model followed by distillation, and end-to-end methods, which train a one-step model from scratch in a single training run. As representatives of the first category, we include the standard diffusion model, following the setup of [14], and Flow Matching, which replaces the diffusion objective with an optimal transport loss as proposed in [9]. These serve as baselines for iterative multi-step denoising models. Several other methods build upon the flow matching objective, often utilizing a teacher model. Reflow [9] is a two-stage distillation technique that generates synthetic (x_0, x_1) pairs by fully evaluating a teacher model. Following [9], 50k synthetic samples are generated for CelebAHQ, with each requiring 128 forward passes. Unlike conventional distillation, the student model is trained across the full time interval $t \in (0, 1)$. Progressive Distillation [16] adopts a binary time-distillation framework. Starting from a pretrained teacher, a sequence of student models is distilled, each trained with a step size that is double the previous one. The initial phase employs classifier-free guidance to enhance performance. Consistency Distillation [20] is a two-stage approach where the student model learns to predict consistent x_1 values from teacher-generated pairs $(x_t, x_{t+\delta})$. In contrast, Consistency Training [20] is an end-to-end method that trains a one-step model directly on empirical pairs $(x_t, x_{t+\delta})$, with time discretization bins increasing progressively during training. Shortcut models [3] propose a novel generative modeling framework that conditions on both the current noise level and desired step size, enabling efficient and flexible sampling under different inference budgets. Finally, Live Reflow, introduced in [3], is an end-to-end model trained

[1]We do not claim that our method outperforms existing distillation techniques; rather, we include some representative distillation methods for reference.

Table 1: Comparison of different training objectives using the same model architecture (**DiT-B**). FID-50k scores (lower is better) are reported for 128, 4, and 1-step denoising. Direct Models produces high-quality samples within a single step and training run, narrowing the gap with single-step distillation methods. Parentheses represent evaluation under conditions that the objective is not intended to support.

	CelebAHQ-256		
	128-Step	**4-Step**	**1-Step**
Distillation			
Progressive Distillation	(302.9)	(251.3)	**14.8**
Consistency Distillation	59.5	39.6	38.2
Reflow	**16.1**	**18.4**	23.2
End-to-end (single training run)			
Diffusion	23.0	(123.4)	(132.2)
Flow Matching	7.3	(63.3)	(280.5)
Consistency Training	53.7	19.0	33.2
Live Reflow	**6.3**	27.2	43.3
Shortcut Models	6.9	**13.8**	20.5
Direct Models (ours)	-	-	**16.8**

Figure 2: Interpolations between two sampled noise points. All displayed images are model generated. Each horizontal set represents images generated by one-step denoising of a variance preserving interpolation between two Gaussian noise samples.

simultaneously on flow-matching and Reflow-distilled targets. The model is conditioned separately on each type of target, and new distillation targets are generated at every training step via full denoising, making the method computationally expensive.

4.3 Evaluation

We follow the evaluation protocol from [3]. Models are evaluated by generating samples using 1 diffusion step for our method, and 128, 4, and 1 steps for the baselines. We report the FID-50k score, a standard metric in generative modeling. FID is computed using statistics from the full dataset, with no compression applied to the generated images. All images are resized to 299×299 using bilinear interpolation and clipped to the $(-1, 1)$ range. During evaluation, we use the Exponential Moving Average (EMA) of the model parameters.

4.4 Results

Table 1 shows that Direct Models achieves strong generation quality using just a single sampling step. Our method outperforms all single-stage training approaches in one-step generation and remains competitive with two-stage progressive distillation. Unlike these multi-phase methods, Direct Models reaches this performance within a single training run. As expected, standard diffusion and flow-matching methods show a significant drop in performance when limited to 4 or 1 sampling step. Additional qualitative results are provided in the supplementary material.

Table 2: Effect of δt on the image quality on CelebAHQ-256.

δt	10^{-2}	$5 \cdot 10^{-3}$
FID ↓	16.8	16.6

4.5 Does Direct Models Induce a Semantically Coherent Latent Space?

To examine whether the latent space learned by Direct Models supports smooth semantic transitions, we explore linear interpolations in the input noise space. Specifically, we select pairs of initial Gaussian noise vectors (x_0^0, x_0^1) and interpolate between them using the variance-preserving formulation

$$x_0^n = n x_0^1 + \sqrt{1 - n^2}\, x_0^0$$

where the coefficient $n \in [0, 1]$. We then pass the interpolated noise samples through the model and observe the corresponding outputs.

Figure 2 presents representative samples from these interpolations. Despite the absence of any explicit constraint or regularizer enforcing smoothness in the learned mapping, the results reveal coherent and continuous transformations across the generated images. These transitions are not only visually smooth but also retain semantic consistency, suggesting that Direct Models constructs a meaningful latent structure.

5 Ablation

We investigate the effect of the discretization interval δt on image quality. As shown in Table 2, both values of δt, namely 10^{-2} and $5 \cdot 10^{-3}$, lead to very comparable FID scores (16.8 vs. 16.6).

6 Related Work

We briefly review existing approaches that enable single-step diffusion-based generation, which can be broadly categorized into distillation-based methods and single-phase training methods.

Distillation Methods In recent years, various techniques have been developed to distill generative models, particularly diffusion models, into more efficient one-step sampling frameworks. These methods typically follow a two-stage pipeline: first, a diffusion model is pretrained; second, a separate model is trained to approximate the behavior of the full diffusion process using fewer inference steps.

Methods such as knowledge distillation [12] and rectified flows [9] generate synthetic training pairs by fully simulating the reverse-time denoising ODE. Due to the high computational cost of full simulation, more efficient alternatives have been proposed that use bootstrapping strategies to partially initialize the ODE trajectory [4, 21]. Additionally, several works have explored alternatives to the standard L2 loss, including adversarial training objectives [17] and distribution-matching approaches [23, 22]. Progressive distillation techniques [16, 1, 13] further decompose the distillation process into multiple stages with increasing time step sizes, thereby reducing the need for long bootstrap paths.

In contrast to these methods, we propose an end-to-end training approach that directly learns a one-step generative model. This eliminates the need for separate pretraining and distillation phases, making our method simpler than both full simulation-based techniques and multi-stage progressive distillation.

Single-phase Training Methods Few methods have been proposed for single-phase training that enable single-step generation. Consistency Models [20], a pioneering approach in this area, represent a class of generative models that directly map partially noised data points to their final, fully denoised outputs in a single step. While these models have been effectively used in distillation purposes, they have also been explored the end-to-end training scenario. Shortcut models [3] propose a novel generative modeling framework that conditions on both the current noise level and desired step size, enabling efficient and flexible sampling under different inference budgets.

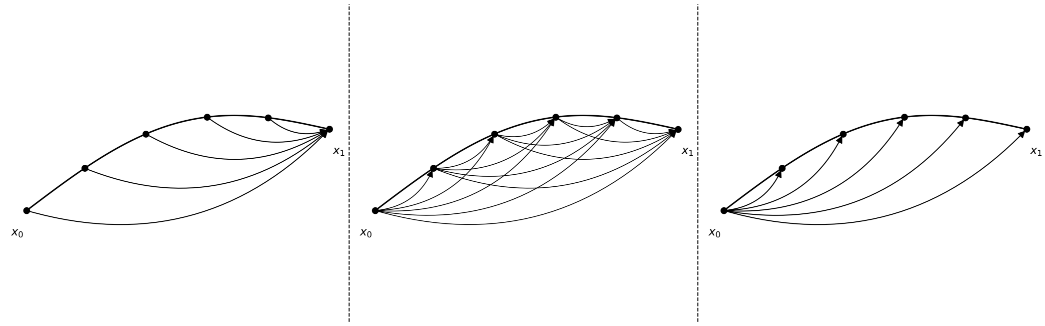

Figure 3: Visual illustration of the differences between prior work in terms of the learned trajectory mappings. x_0 denotes an initial Gaussian noise and x_1 its corresponding noise-free image. Left: Consistency models [20] . Middle: Shortcuts models [3]. Right: Direct Models (ours).

Difference from consistency models [20] While we share the general concept of consistency with [20], Direct Models differs significantly in its formulation. Conceptually, Direct Models can be seen as the opposite approach: instead of mapping intermediate latents x_t directly to the fully denoised image x_1, our method maps initial Gaussian noise x_0 to all intermediate latent states x_t, as shown in Figure 3.

Difference from shortcut models [3] Shortcut models are arguably the most similar approach to Direct Models. However, there are key differences: 1) Conceptually, Shortcut models learn direct mappings between all pairs of latent states, including both the initial and final ones. In contrast, Direct Models focuses on learning a direct mapping only between the initial Gaussian noise and all intermediate latent states. 2) More importantly, our approach is grounded on a more principled mathematical formulation. Specifically, we derive our method using a Taylor expansion of the flow, as shown in Equation (13), which links the flow at neighboring timesteps to the velocity field.

7 Limitations and Future Work

Our method presents some limitations. First, Direct Models requires training two separate networks, which limits efficiency. Second, the current formulation is restricted to single-step inference. A promising direction for future research is to extend our framework to enable training a single unified model while potentially allowing flexibility in the number of inference steps during sampling.

8 Conclusion

We introduced Direct Models, a new type of diffusion-based generative model that enables both single-step sampling and single-run training. By learning a time-indexed residual field to directly approximate the full generative flow, our method achieves fast and high-quality generation while significantly simplifying the training process. This makes Direct Models a practical and efficient alternative to existing diffusion-based techniques.

References

[1] 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. 7

[2] 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. 5

[3] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024. 5, 6, 7, 8

[4] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference $\{\backslash\&\}$ Generative Modeling*, 2023. 7

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[6] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023. 2

[7] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 1

[8] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[9] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2, 5, 7

[10] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 1

[11] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 5

[12] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 1, 7

[13] 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. 7

[14] 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. 4, 5

[15] 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. 5

[16] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 1, 5, 7

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

[18] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015. 2

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

[20] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023. 5, 7, 8

[21] 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, 2024. *URL https://arxiv. org/abs/2405.16852*. 7

[22] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024. 7

[23] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024. 7

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: the claims made reflect the paper contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a Limitations section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We included the assumptions and the full derivations.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will share our code in the supplementary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: we will share the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We will share all the details in the supplementary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow prior works that do not .

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We will provide this information if the reviewers ask for it.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We believe there is no major societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no high risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the used assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer:[NA]

Justification: No new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowd-sourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No risks.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: No major use of LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.