Stable Consistency Tuning: Understanding and Improving Consistency Models

Fu-Yun Wang
MMLab, CUHK
Hong Kong SAR
fywang@link.cuhk.edu.hk

Zhengyang Geng
Carnegie Mellon University
Pittsburgh, USA
zhengyanggeng@gmail.com

Hongsheng Li
MMLab, CUHK
Hong Kong SAR
hsli@ee.cuhk.edu.hk

Abstract

Diffusion models achieve high-quality generation but suffer from slow sampling due to their iterative denoising process. Consistency models offer a faster alternative with competitive performance, trained via consistency distillation from pretrained diffusion models or directly from raw data. We introduce a novel framework interpreting consistency models through a Markov Decision Process (MDP), framing their training as value estimation via Temporal Difference (TD) Learning. This perspective reveals limitations in existing training strategies. Building on Easy Consistency Tuning (ECT), we propose Stable Consistency Tuning (SCT), which enhances variance reduction using the score identity. SCT significantly improves performance on CIFAR-10 and ImageNet-64.

1 Introduction

Diffusion models have achieved state-of-the-art performance in generating images (Dhariwal & Nichol, 2021; Rombach et al., 2022; Song & Ermon, 2019; Karras et al., 2022; 2024; Wang et al., 2025), videos (Shi et al., 2024; Blattmann et al., 2023; Singer et al., 2022; Brooks et al., 2024; Bao et al., 2024; Wang et al., 2023; 2024d;b; Bian et al., 2025), 3D (Gao et al., 2024; Shi et al., 2023; Lai et al., 2025), and 4D data (Ling et al., 2024). These models iteratively refine noise into clean samples, yielding high-quality results with stable training (Goodfellow et al., 2020; Sauer et al., 2023a). However, their reliance on iterative inference leads to high computational costs (Song et al., 2020; Ho et al., 2020), making practical applications such as real-time generation challenging, especially for high-resolution images and videos. Consistency models (Song et al., 2023) address these limitations by enabling high-quality, one-step generation without adversarial training. Recent studies (Song & Dhariwal, 2023; Geng et al., 2024) show that their one-step and two-step performance can rival diffusion models, which require significantly more inference steps. These models enforce the self-consistency condition (Song et al., 2023) along probability flow ODE (PF-ODE) trajectories through two main training methods: consistency distillation (CD) and consistency training/tuning (CT). CD utilizes a pretrained diffusion model to simulate the PF-ODE, while CT learns directly from real data.

This work provides a unified understanding of consistency models through the lens of bootstrapping. We frame the reverse diffusion process as a Markov Decision Process (MDP), aligning consistency training with Temporal Difference (TD) learning (Sutton & Barto, 2018). Our analysis reveals that CD has lower variance and greater stability but is constrained by the pretrained model, while CT offers higher potential but suffers from instability due to reward estimation variance. To address this, we introduce **Stable Consistency Tuning (SCT)**, which leverages variance-reduced training via the score identity (Vincent, 2011; Xu et al., 2023) and a smoother progressive training schedule, enhancing both stability and performance. Our core motivation is shown in Fig. 1.

2 Preliminaries

Diffusion models generate data by solving a probability flow ordinary differential equation (PF-ODE), which describes the deterministic evolution of data samples along a learned trajectory. These models typically rely on a neural network to approximate the score function, enabling numerical solvers

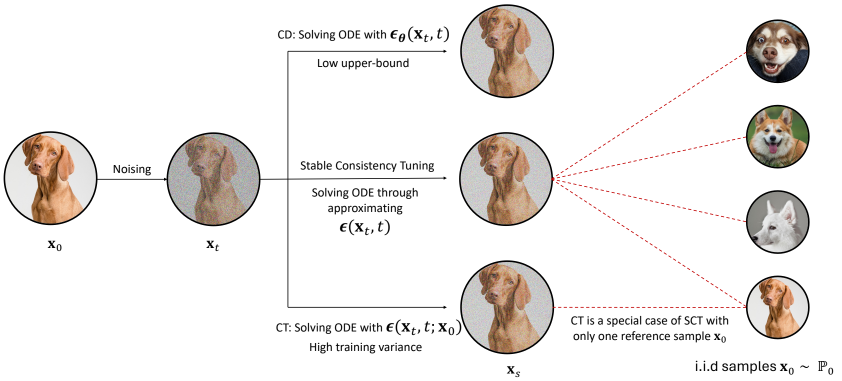

CD: Solving ODE with $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$
Low upper-bound
Noising
Stable Consistency Tuning
Solving ODE through approximating $\boldsymbol{\epsilon}(\mathbf{x}_t, t)$
$\mathbf{x}_0$
$\mathbf{x}_t$
CT: Solving ODE with $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0)$
High training variance
$\mathbf{x}_s$
CT is a special case of SCT with only one reference sample $\mathbf{x}_0$
i.i.d samples $\mathbf{x}_0 \sim \mathbb{P}_0$

Figure 1: Stable consistency tuning (SCT) with variance reduced training target. SCT provides a unifying perspective to understand different training strategies of consistency models.

to generate samples efficiently. **Consistency models** learn to directly map noisy samples to clean data in a single step by predicting the solution of the PF-ODE, satisfying $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \mathbf{x}_0, \forall t \in [0, 1]$. They are trained using a consistency loss, $d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x}_r, r))$, which ensures different noisy samples along the same trajectory produce consistent outputs, enabling efficient and high-quality data generation.

3 Stable Consistency Tuning

3.1 Understanding consistency models

For a general form of diffusion $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$, the consistency model aims to learn a $\mathbf{x}_0$ predictor with only the information from $\mathbf{x}_t, \forall t \in [0, 1]$. Specifically, for consistency models formulated as $\hat{\mathbf{x}}_0(\mathbf{x}_t, t; \boldsymbol{\theta}) = \frac{1}{\alpha_t}\mathbf{x}_t - \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$, where $\boldsymbol{h}_{\boldsymbol{\theta}}$ approximates the weighted integral of $\boldsymbol{\epsilon}$, we have the self-consistency learning objective:

$$\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \xleftarrow{\text{fit}} \boldsymbol{r} + \boldsymbol{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_r, r)\,, \tag{1}$$

where $\boldsymbol{r}$ represents the integral of $\boldsymbol{\epsilon}$ over the time interval. This formulation aligns with the Bellman equation, with $\boldsymbol{h}_{\boldsymbol{\theta}}$ serving as a value function. Particularly, we show the diffusion process can be modeled as a Markov Decision Process (MDP) and consistency training corresponds to a value estimation problem in Temporal Difference Learning (TD-Learning). We provide a detailed discussion in Section II.

3.2 Reducing the Training Variance

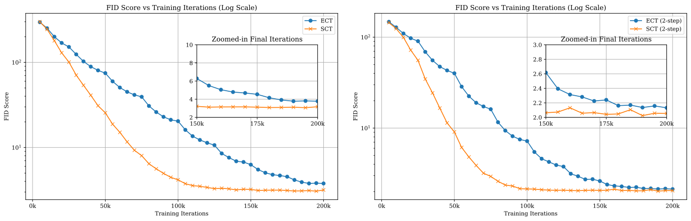
Previous research has shown that reducing the variance for diffusion training can lead to improved training stability and performance (Xu et al., 2023). However, this technique has only been applied to unconditional generation and diffusion model training. We generalize this technique to both conditional/unconditional generation and consistency training/tuning for variance reduction. Let $\boldsymbol{c}$ represent the conditional inputs (*e.g.*, class labels). We show the conditional epsilon estimation adopted in consistency training/tuning can be replaced by our variance-reduced estimation in Theorem 1:

$$\boldsymbol{\epsilon}(\mathbf{x}_t, t) = -\sigma_t \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t) = \frac{1}{n}\sum_{i=0}^{n-1} W_i \boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)}))\,, \tag{2}$$

where $W_i = \frac{\mathbb{P}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t|\mathbf{x}_0^{(j)})}$ is the weight of conditional $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})$. In simple terms, other data from the dataset can be used to adjust the learning objective of the consistency model, reducing training variance and enhancing stability, as illustrated in Fig. 1.

3.3 Reducing the Discretization Error

To achieve higher performance, we minimize $\Delta t = (t - r)$. A large Δt increases discretization errors, while a small Δt may cause error accumulation or training failure. Previous works (Song



(a) 1-step

(b) 2-step

Figure 2: FID vs Training iterations. SCT has faster convergence speed and better performance upper bound than ECT.

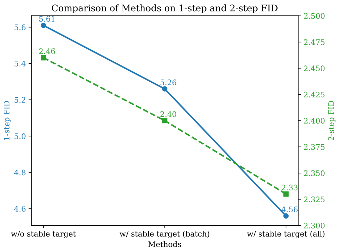


Figure 3: The effectiveness of variance reduced training target.

et al., 2023; Song & Dhariwal, 2023; Geng et al., 2024) adopt progressive training, starting with a large Δt and gradually reducing it. This accelerates optimization initially and later refines results, improving performance. The ECT training schedule follows:

$$t \sim \text{LogNormal}(P_{\text{mean}}, P_{\text{std}}), \quad r := \text{ReLU}\left(1 - \frac{1}{q^{\lceil \text{iter}/d \rceil}} n(t)\right) t\,, \tag{3}$$

where q controls shrinking speed, d determines frequency, and ReLU is $\max(\cdot, 0)$. We empirically find that a smoother shrinking process is beneficial, achieved by reducing both q and d, leading to faster and more stable training. Following (Song & Dhariwal, 2023; Geng et al., 2024), we apply weighting $1/(t - r)$. Assuming $r = \alpha t$, the weighting decomposes into $\frac{1}{t} \times \frac{1}{(1-\alpha)}$. The term $1/t$ prioritizes smaller timesteps, ensuring stable predictions as teachers for larger steps. The factor $1/(1 - \alpha)$ increases weight as Δt shrinks, preventing gradient vanishing. To avoid instability when Δt is too small, we introduce a smooth term $\delta > 0$ in the weighting function: $\frac{1}{t-r+\delta} \leqslant \frac{1}{\delta}$.

4 Experiments

4.1 Experiment Setups

Evaluation Benchmarks. Following the evaluation protocols of iCT (Song & Dhariwal, 2023) and ECT (Geng et al., 2024), we validate the effectiveness of SCT on CIFAR-10 (unconditional) (Krizhevsky et al., 2009) and ImageNet-64 (conditional) (Deng et al., 2009). Performance is measured using Frechet Inception Distance (FID, lower is better) (Heusel et al., 2017) consistent with recent studies (Geng et al., 2024; Karras et al., 2024).

Compared baselines. We compare our method against accelerated samplers (Lu et al., 2022; Zhao et al., 2024), state-of-the-art diffusion-based methods (Ho et al., 2020; Song & Ermon, 2019; 2020; Karras et al., 2022), distillation methods (Zhou et al., 2024; Salimans & Ho, 2022), alongside consistency training and tuning approaches. Among these models, consistency training and tuning methods serve as key baselines, including CT (LPIPS) (Song et al., 2023), iCT (Song & Dhariwal, 2023), ECT (Geng et al., 2024), and MCM (CT) (Heek et al., 2024). From a model perspective, iCT is based on the ADM (Dhariwal & Nichol, 2021), ECT is built on EDM2 (Karras et al., 2024), and MCM follows the UViTs of Simple Diffusion (Hoogeboom et al., 2023). The model size of ECT is similar to that of iCT, while MCM does not explicitly specify the model size. The iCT model is randomly initialized, whereas both ECT and MCM use pretrained diffusion models for initialization. In terms of training costs, iCT uses a batch size of 4096 across 800,000 iterations, MCM employs a batch size of 2048 for 200,000 iterations, and ECT utilizes a batch size of 128 for 100,000 iterations. SCT follows ECT's model architecture and training configuration.

4.2 Results and Analysis

Training efficiency and efficacy. In Fig. 2, we plot 1-step FID and 2-step FID for SCT and ECT along the number of training epochs, under the same training configuration. From the figure, we observe that SCT significantly improves convergence speed compared to ECT, demonstrating the efficiency and efficacy of SCT training. Additionally, the performance comparisons in Tables 1 and 2 also show that SCT outperforms ECT across different settings.

Table 1: Comparison on CIFAR-10.

METHOD	NFE (↓)	FID (↓)
Fast samplers & distillation for diffusion models		
DDIM (Song et al., 2020)	10	13.36
DPM-solver-fast (Lu et al., 2022)	10	4.70
3-DEIS (Zhang & Chen, 2022)	10	4.17
UniPC (Zhao et al., 2024)	10	3.87
Knowledge Distillation (Luhman & Luhman, 2021)	1	9.36
DFNO (LPIPS) (Zheng et al., 2022)	1	3.78
2-Rectified Flow (+distill) (Liu et al., 2022)	1	4.85
TRACT (Berthelot et al., 2023)	1	3.78
Diff-Instruct (Luo et al., 2023)	1	4.53
PD (Salimans & Ho, 2022)	1	8.34
	2	5.58
CTM (Kim et al., 2023)	1	5.19
	18	3.00
CTM (+GAN +CRJ)	1	1.98
	2	1.87
SiD ($\alpha = 1.0$) (Zhou et al., 2024)	1	2.03
SiD ($\alpha = 1.2$) (Zhou et al., 2024)	1	1.98
CD (LPIPS) (Song et al., 2023)	1	3.55
	2	2.93
Direct Generation		
Score SDE (Song et al., 2021)	2000	2.38
Score SDE (deep) (Song et al., 2021)	2000	2.20
DDPM (Ho et al., 2020)	1000	3.17
LSGM (Vahdat et al., 2021)	147	2.10
PFGM (Xu et al., 2022)	110	2.35
EDM (Karras et al., 2022)	35	2.04
NVAE (Vahdat & Kautz, 2020)	1	23.5
BigGAN (Brock et al., 2019)	1	14.7
StyleGAN2 (Karras et al., 2020)	1	8.32
Consistency Training/Tuning		
CT (LPIPS) (Song et al., 2023)	1	8.70
	2	5.83
iCT (Song & Dhariwal, 2023)	1	2.83
	2	2.46
iCT-deep (Song & Dhariwal, 2023)	1	2.51
	2	2.24
ECT (Geng et al., 2024)	1	3.78
	2	2.13
SCT	1	3.11
	2	2.05

Table 2: Comparison on ImageNet-64.

METHOD	NFE (↓)	FID (↓)
Fast samplers & distillation for diffusion models		
DDIM (Song et al., 2020)	50	13.7
	10	18.3
DPM solver (Lu et al., 2022)	10	7.93
	20	3.42
DEIS (Zhang & Chen, 2022)	10	6.65
	20	3.10
DFNO (LPIPS) (Zheng et al., 2022)	1	7.83
TRACT (Berthelot et al., 2023)	1	7.43
	2	4.97
BOOT (Gu et al., 2023)	1	16.3
Diff-Instruct (Luo et al., 2023)	1	5.57
PD (Salimans & Ho, 2022)	1	15.39
	2	8.95
CTM (+GAN + CRJ) (Kim et al., 2023)	1	1.92
SID ($\alpha = 1.0$) (Zhou et al., 2024)	1	2.03
PD (LPIPS) (Song et al., 2023)	1	7.88
	2	5.74
CD (LPIPS) (Song et al., 2023)	1	6.20
	2	4.70
Direct Generation		
RIN (Jabri et al., 2022)	1000	1.23
DDPM (Ho et al., 2020)	250	11.0
iDDPM (Nichol & Dhariwal, 2021)	250	2.92
ADM (Dhariwal & Nichol, 2021)	250	2.07
EDM (Karras et al., 2022)	511	1.36
EDM* (Heun) (Karras et al., 2022)	79	2.44
BigGAN-deep (Brock et al., 2019)	1	4.06
Consistency Training/Tuning		
CT (LPIPS) (Song et al., 2023)	1	13.0
	2	11.1
iCT (Song & Dhariwal, 2023)	1	4.02
	2	3.20
iCT-deep (Song & Dhariwal, 2023)	1	3.25
	2	2.77
MCM (CT) (Heek et al., 2024)	1	7.2
	2	2.7
ECT-M (Geng et al., 2024)	1	3.67
	2	2.35
SCT-M	1	3.30
	2	2.13

Results for existing methods are taken from previous papers. Results of SCT on CIFAR-10 are trained with batch size 128 for 200k iterations. Results of SCT on ImageNet-64 are trained with batch size 128 for 100k iterations.

Quantitative evaluation. We present results in Table 1 and Table 2. Our approach consistently outperforms ECT across various scenarios, achieving results comparable to advanced distillation strategies and diffusion/score-based models.

The effectiveness of training variance reduction. It is worth noting that SCT and ECT employ different progressive training schedules. To exclude this effect, we adopt ECT's fixed training schedule, in which the 2-step FID surpasses Consistency Distillation within a single A100 GPU hour. We use $\Delta t = t/256$ as a fixed partition, with a batch size of 128, over 16k iterations on CIFAR-10, while keeping all other settings unchanged. For SCT models on CIFAR-10, we calculate the variance-reduced target only within the training batch, which is also the default setting of all our experiments on CIFAR-10. To further showcase the effectiveness of the variance-reduced target, we use all 50,000 training samples as a reference to compute the target. Although more reference samples are used, they do not directly influence the model's computations; they are solely utilized for calculating the training target. Fig. 3 presents a comparison of these three methods, showing that our approach achieves notable improvements in both 1-step and 2-step FID. Notably, when using the entire sample set as the reference batch, the improvement becomes more pronounced, with the 1-step FID dropping from 5.61 to 4.56.

5 Conclusion and Limitations

We propose Stable Consistency Tuning (SCT), a unified approach that improves consistency models by reducing training variance and discretization errors. SCT achieves faster convergence and state-of-the-art generative performance in 1-step and few-step sampling on CIFAR-10 and ImageNet-64×64. While our work focuses on unconditional and class-conditional generation on standard benchmarks, similar to iCT and ECT, future research should explore consistency training/tuning at larger scales or more advanced settings such as text-to-image generation.

References

Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

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.

Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. Gs-dit: Advancing video generation with pseudo 4d gaussian fields through efficient dense 3d point tracking. *arXiv preprint arXiv:2501.02690*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Ricky TQ Chen and Yaron Lipman. Riemannian flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

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

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy, 2024. URL https://arxiv.org/abs/2406.14548.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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 {\&} Generative Modeling*, 2023.

Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

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. *arXiv preprint arXiv:2310.02279*, 2023.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. On density estimation with diffusion models. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *NeurIPS*, 2021. URL https://openreview.net/forum?id=2LdBqxc1Yv.

Fei Kong, Jinhao Duan, Lichao Sun, Hao Cheng, Renjing Xu, Hengtao Shen, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. Act-diffusion: Efficient adversarial consistency training for one-step diffusion models, 2024.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. Cllms: Consistency large language models, 2024.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qinxiang Lin, Jinwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8576–8588, 2024.

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

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.

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

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

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *arXiv preprint arXiv:2305.18455*, 2023.

Xiaofeng Mao, Zhengkai Jiang, Fu-Yun Wang, Wenbing Zhu, Jiangning Zhang, Hao Chen, Mingmin Chi, and Yabiao Wang. Osv: One step is enough for high-quality image to video generation. *arXiv preprint arXiv:2409.11367*, 2024.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, pp. 14297–14306, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, October 2023.

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. *arXiv preprint arXiv:2307.01952*, 2023.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation, 2024.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

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

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

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023b.

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

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

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11918–11930, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Advances in neural information processing systems*, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. URL https://api.semanticscholar.org/CorpusID:5560643.

Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*, 2023.

Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 2024a.

Fu-Yun Wang, Zhaoyang Huang, Qiang Ma, Guanglu Song, Xudong Lu, Weikang Bian, Yijin Li, Yu Liu, and Hongsheng Li. Zola: Zero-shot creative long animation generation with short video model. In *European Conference on Computer Vision*, pp. 329–345. Springer, 2024b.

Fu-Yun Wang, Zhaoyang Huang, Xiaoyu Shi, Weikang Bian, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning. *arXiv preprint arXiv:2402.00769*, 2024c.

Fu-Yun Wang, Xiaoshi Wu, Zhaoyang Huang, Xiaoyu Shi, Dazhong Shen, Guanglu Song, Yu Liu, and Hongsheng Li. Be-your-outpainter: Mastering video outpainting through input-specific adaptation. In *European Conference on Computer Vision*, pp. 153–168. Springer, 2024d.

Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. *arXiv preprint arXiv:2410.07303*, 2024e.

Fu-Yun Wang, Yunhao Shui, Jingtan Piao, Keqiang Sun, and Hongsheng Li. Diffusion-npo: Negative preference optimization for better preference aligned generation of diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi S. Jaakkola. Poisson flow generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=voV_TRqcWh.

Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models. *arXiv preprint arXiv:2302.00670*, 2023.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. *arXiv preprint arXiv:2211.13449*, 2022.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

APPENDIX

I RELATED WORKS

Diffusion Models. Diffusion models (Ho et al., 2020; Song et al., 2021; Karras et al., 2022) have emerged as leading foundational models in image synthesis. Recent studies have developed their theoretical foundations (Lipman et al., 2022; Chen & Lipman, 2023; Song et al., 2021; Kingma et al., 2021) and sought to expand and improve the sampling and design space of these models (Song et al., 2020; Karras et al., 2022; Kingma et al., 2021). Other research has explored architectural innovations for diffusion models (Dhariwal & Nichol, 2021; Peebles & Xie, 2023), while some have focused on scaling these models for text-conditioned image synthesis and various real-world applications (Shi et al., 2024; Rombach et al., 2022; Podell et al., 2023). Efforts to accelerate the sampling process include approaches at the scheduler level (Karras et al., 2022; Lu et al., 2022; Song et al., 2020) and the training level (Meng et al., 2023; Song et al., 2023), with the former often aiming to improve the approximation of the probability flow ODE (Lu et al., 2022; Song et al., 2020). The latter primarily involves distillation techniques (Meng et al., 2023; Salimans & Ho, 2022; Wang et al., 2024e) or initializing diffusion model weights for GAN training (Sauer et al., 2023b; Lin et al., 2024).

Consistency Models. Consistency models are an emerging class of generative models (Song et al., 2023; Song & Dhariwal, 2023) for fast high-quality generation. It can be trained through either consistency distillation or consistency training. Advanced methods have demonstrated that consistency training can surpass diffusion model training in performance (Song & Dhariwal, 2023; Geng et al., 2024). Several studies propose different strategies for segmenting the ODE (Kim et al., 2023; Heek et al., 2024; Wang et al., 2024a), while others explore combining consistency training with GANs to enhance training efficiency (Kong et al., 2024). Additionally, the consistency model framework has been applied to video generation (Wang et al., 2024c; Mao et al., 2024), language modeling (Kou et al., 2024) and policy learning (Prasad et al., 2024).

II UNDERSTANDING CONSISTENCY MODELS

Consistency model as bootstrapping. For a general form of diffusion $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$, there exists an exact solution form of PF-ODE as shown in previous work (Song et al., 2021; Lu et al., 2022),

$$\mathbf{x}_s = \frac{\alpha_s}{\alpha_t}\mathbf{x}_t - \alpha_s \int_{\lambda_t}^{\lambda_s} e^{-\lambda} \boldsymbol{\epsilon}(\mathbf{x}_{t_\lambda}, t_\lambda) \mathrm{d}\lambda \,, \tag{4}$$

where $\lambda_t = \ln(\alpha_t/\sigma_t)$, λ_t is the reverse function of t_λ, and $\boldsymbol{\epsilon}(\mathbf{x}_{t_\lambda}, t_\lambda) = -\sigma_{t_\lambda} \nabla \log \mathbb{P}_{t_\lambda}(\mathbf{x}_{t_\lambda})$ is the scaled score function. Consistency models aim to learn a $\mathbf{x}_0$ predictor with only the information from $\mathbf{x}_t, \forall t \in [0,1]$. The left term is already known with $\mathbf{x}_t$, and thereby we can write the consistency model-based $\mathbf{x}_0$ prediction as

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t; \boldsymbol{\theta}) = \frac{1}{\alpha_t}\mathbf{x}_t - \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tag{5}$$

where s is set to 0 with $\alpha_s = 1$, $\boldsymbol{\theta}$ is the model weights, and $\boldsymbol{h}_{\boldsymbol{\theta}}$ is applied to approximate the weighted integral of $\boldsymbol{\epsilon}$ from t to $s = 0$.

The loss of consistency models penalize the $\mathbf{x}_0$ prediction distance between $\mathbf{x}_t$ and $\mathbf{x}_r$ at adjacent timesteps,

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t; \boldsymbol{\theta}) \xleftarrow{\text{fit}} \hat{\mathbf{x}}_0(\mathbf{x}_r, r; \boldsymbol{\theta}^-) \,, \tag{6}$$

where $0 \leqslant r < t$ are timesteps and $\boldsymbol{\theta}^-$ is the EMA weight of $\boldsymbol{\theta}$. Therefore, we have the following learning target

$$\frac{1}{\alpha_t}\mathbf{x}_t - \boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \xleftarrow{\text{fit}} \frac{1}{\alpha_r}\mathbf{x}_r - \boldsymbol{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_r, r) \tag{7}$$

Noting that $\mathbf{x}_r = \frac{\alpha_r}{\alpha_t}\mathbf{x}_t - \alpha_r \int_{\lambda_t}^{\lambda_r} e^{-\lambda} \boldsymbol{\epsilon}(\mathbf{x}_{t_\lambda}, t_\lambda) \mathrm{d}\lambda$, and hence we replace the $\mathbf{x}_r$ in the above equation and have

$$\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \xleftarrow{\text{fit}} \boldsymbol{r} + \boldsymbol{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_r, r) \,, \tag{8}$$

Table 4: The definition of symbols in the value estimation of the PF-ODE equivalent MDP.

MDP symbols	Definition	MDP symbols	Definition
s_{t_n}	$(t_{N-n}, \mathbf{x}_{t_{N-n}})$	a_{t_n}	$\mathbf{x}_{t_{N-n-1}} := \Phi(\mathbf{x}_{t_{N-n}}, t_{N-n}, t_{N-n-1})$
$P_0(s_0)$	$(t_N, \mathcal{N}(\mathbf{0}, \mathbf{I}))$	$P(s_{t_{n+1}} \mid s_{t_n}, a_{t_n})$	$(\delta_{t_{N-n-1}}, \delta_{\mathbf{x}_{t_{N-n-1}}})$
$\pi(a_{t_n} \mid s_{t_n})$	$\delta_{\mathbf{x}_{t_{N-n-1}}}$	$R(s_{t_n}, a_{t_n})$	$\int_{\lambda_{t_{N-n}}}^{\lambda_{t_{N-n-1}}} e^{-\lambda} \boldsymbol{\epsilon}(\mathbf{x}_{t_\lambda}, t_\lambda) \mathrm{d}\lambda$
$V_{\boldsymbol{\theta}}(s_{t_n})$	$\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{N-n}}, t_{N-n})$		

where $\boldsymbol{r} = \int_{\lambda_t}^{\lambda_r} e^{-\lambda} \boldsymbol{\epsilon}(\mathbf{x}_{t_\lambda}, t_\lambda) \mathrm{d}\lambda$. The above equation is a Bellman Equation. $\boldsymbol{h}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ is the value estimation at state $\mathbf{x}_t$, $\boldsymbol{h}_{\boldsymbol{\theta}^-}(\mathbf{x}_r, r)$ is the value estimation at state $\mathbf{x}_r$, and $\boldsymbol{r}$ is the step 'reward'.

Standard formulation. It is known that the diffusion generation process can be modeled as a Markov Decision Process (MDP) (Black et al., 2023; Fan et al., 2024), and here we show that the training of consistency models can be viewed as a value estimation learning process, which is also known as Temporal Difference Learning (TD-Learning), in the equivalent MDP. We show the standard formulation in Table 4. In Table 4, s_{t_n} and a_{t_n} are the state and action at timestep t_n, P_0 and P are the initial state distribution and state transition distribution, $\Phi(\mathbf{x}_{t_{N-n}}, t_{N-n}, t_{N-n-1})$ is the ODE solver, π is the policy following the PF-ODE, reward R is equivalent to the r defined above and value function $V_{\boldsymbol{\theta}}$ is corresponding to $\boldsymbol{h}_{\boldsymbol{\theta}}$. π is the Dirac distribution δ due to the deterministic nature of PF-ODE. From this perspective, we can have a unifying understanding of consistency model variants and their behaviors. Fig. 1 provides a straightforward illustration of our insight. One of the most important factors of the consistency model performance is how we estimate $\boldsymbol{r}$ in the equation.

III Proofs

Lemma 1. *The ground truth $\boldsymbol{\epsilon}(\mathbf{x}_t, t)$ is the expectation value of conditional epsilon prediction $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0)$ over the distribution $\mathbb{P}(\mathbf{x}_0 \mid \mathbf{x}_t)$,* i.e., $\boldsymbol{\epsilon}(\mathbf{x}_t, t) = \mathbb{E}_{\mathbb{P}(\mathbf{x}_0|\mathbf{x}_t)}\left[\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0)\right]$.

Proof.

$$\begin{aligned}
\boldsymbol{\epsilon}(\mathbf{x}_t, t) &= -\sigma_t \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t) \\
&= -\sigma_t \mathbb{E}_{\mathbb{P}_t(\mathbf{x}_0|\mathbf{x}_t)}\left[\nabla_{\mathbf{x}_t} \log \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0)\right] \\
&= -\sigma_t \mathbb{E}_{\mathbb{P}(\mathbf{x}_0|\mathbf{x}_t)}\left[-\frac{\mathbf{x}_t - \alpha_t \mathbf{x}_0}{\sigma_t^2}\right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0|\mathbf{x}_t)}\left[\frac{\mathbf{x}_t - \alpha_t \mathbf{x}_0}{\sigma_t}\right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0|\mathbf{x}_t)}\left[\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0)\right]
\end{aligned} \quad , \tag{9}$$

□

Theorem 1. *We show the ground truth epsilon estimation can be approximated by numerical computation with sampled training data,* i.e.,

$$\boldsymbol{\epsilon}(\mathbf{x}_t, t) = -\sigma_t \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t) = \frac{1}{n} \sum_{i=0}^{n-1} W_i \boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})) \, , \tag{10}$$

where $W_i = \frac{\mathbb{P}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t|\mathbf{x}_0^{(j)})}$ is the weight of conditional $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})$.

Proof.

$$
\begin{aligned}
\nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \boldsymbol{c}) &= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0 \mid \mathbf{x}_t, \boldsymbol{c})} \left[\nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{c}) \right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0 \mid \boldsymbol{c})} \left[\frac{\mathbb{P}(\mathbf{x}_0 \mid \mathbf{x}_t, \boldsymbol{c})}{\mathbb{P}(\mathbf{x}_0 \mid \boldsymbol{c})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{c}) \right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0 \mid \boldsymbol{c})} \left[\frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{c})}{\mathbb{P}(\mathbf{x}_t \mid \boldsymbol{c})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0, \boldsymbol{c}) \right] \\
&= \mathbb{E}_{\mathbb{P}(\mathbf{x}_0 \mid \boldsymbol{c})} \left[\frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0)}{\mathbb{P}(\mathbf{x}_t \mid \boldsymbol{c})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0) \right] \\
&\approx \frac{1}{n} \sum_{\{\mathbf{x}_0^{(i)}\} \sim \mathbb{P}(\mathbf{x}_0 \mid c)}^{i=0,\dots n-1} \frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})}{\mathbb{P}(\mathbf{x}_t \mid \boldsymbol{c})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0^{(i)}) \qquad (11) \\
&\approx \frac{1}{n} \sum_{\{\mathbf{x}_0^{(i)}\} \sim \mathbb{P}(\mathbf{x}_0 \mid c)}^{i=0,\dots n-1} \frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(j)}, \boldsymbol{c})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0^{(i)}) \\
&= \frac{1}{n} \sum_{\{\mathbf{x}_0^{(i)}\} \sim \mathbb{P}(\mathbf{x}_0 \mid c)}^{i=0,\dots n-1} \frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(j)})} \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})
\end{aligned}
$$

The key difference between the variance-reduced score estimation of conditional generation and unconditional generation is whether the samples utilized for computing the variance-reduced target are sampled from the conditional distribution $\mathbb{P}(\mathbf{x}_0 \mid \boldsymbol{c})$ or not. In the class-conditional generation, this means we compute stable targets only within each class cluster. For text-to-image generation, we might estimate probabilities using CLIP (Radford et al., 2021) text-image similarity, though we leave this for future study.

Then, according to Lemma 1, we can easily observe that

$$\boldsymbol{\epsilon}(\mathbf{x}_t, t) = -\sigma_t \nabla_{\mathbf{x}_t} \log \mathbb{P}_t(\mathbf{x}_t) = \frac{1}{n} \sum_{i=0}^{n-1} W_i \boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})) \, . \qquad (12)$$

where $W_i = \frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(j)})}$ is the weight of conditional $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})$. We proceed by explicitly computing the weight W_i. Given that the transition probability follows a high-dimensional Gaussian distribution:

$$\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0) = \frac{1}{(2\pi\sigma_t^2)^{d/2}} \exp\left(-\frac{1}{2\sigma_t^2} \|\mathbf{x}_t - \mathbf{x}_0\|^2 \right), \qquad (13)$$

the weight W_i is defined as:

$$W_i = \frac{\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(i)})}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0^{(j)})}. \qquad (14)$$

Substituting the Gaussian density function, we obtain:

$$W_i = \frac{\exp\left(-\frac{1}{2\sigma_t^2} \|\mathbf{x}_t - \mathbf{x}_0^{(i)}\|^2 \right)}{\sum_{\mathbf{x}_0^{(j)} \in \{\mathbf{x}_0^{(i)}\}} \exp\left(-\frac{1}{2\sigma_t^2} \|\mathbf{x}_t - \mathbf{x}_0^{(j)}\|^2 \right)}. \qquad (15)$$

This formulation shows that W_i is a softmax function over the negative squared Euclidean distances between $\mathbf{x}_t$ and different $\mathbf{x}_0^{(j)}$, scaled by σ_t^2. This weight determines the contribution of the conditional noise term $\boldsymbol{\epsilon}(\mathbf{x}_t, t; \mathbf{x}_0^{(i)})$ relative to all possible values $\mathbf{x}_0^{(j)}$. The softmax structure implies that the weight is higher for values of $\mathbf{x}_0^{(j)}$ that are closer to $\mathbf{x}_t$, with the scaling controlled by σ_t^2. □

IV Qualitative Results

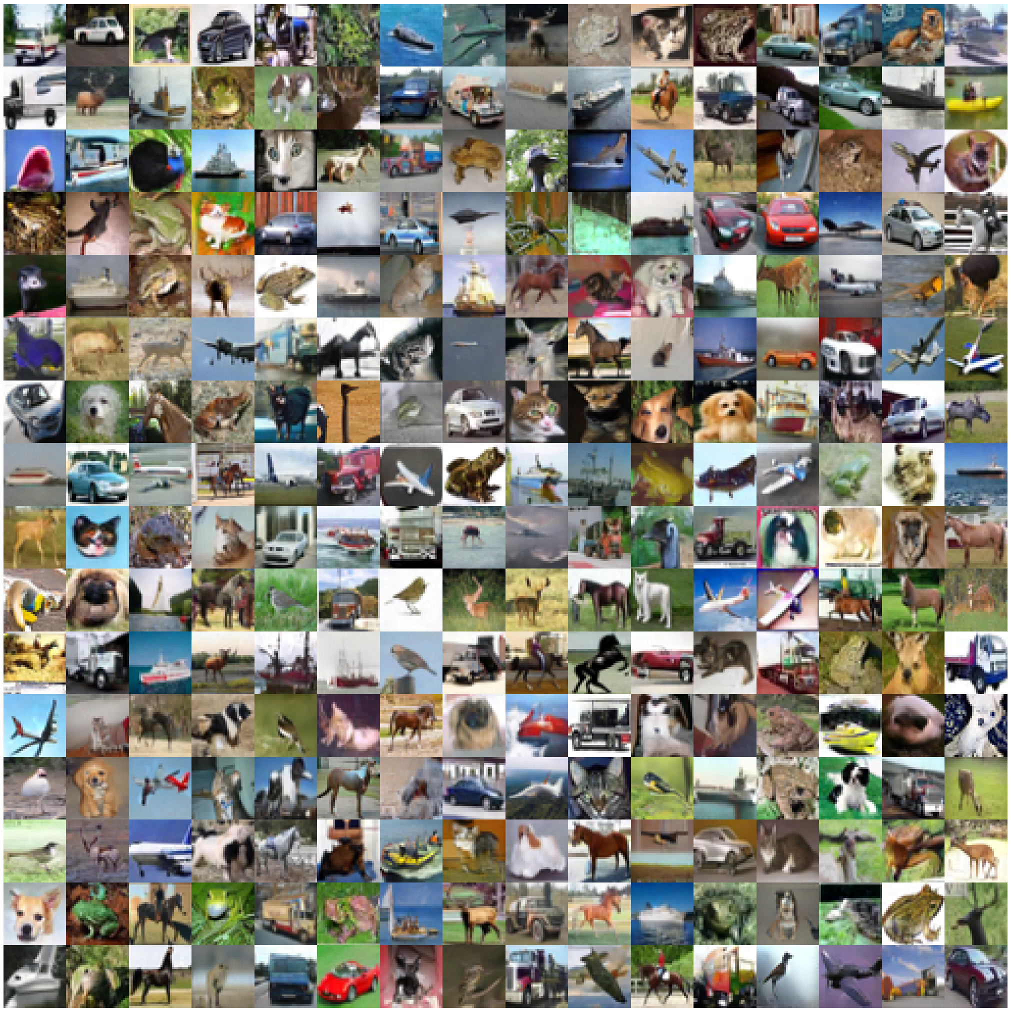


Figure 4: 1-step samples from unconditional SCT trained on CIFAR-10.

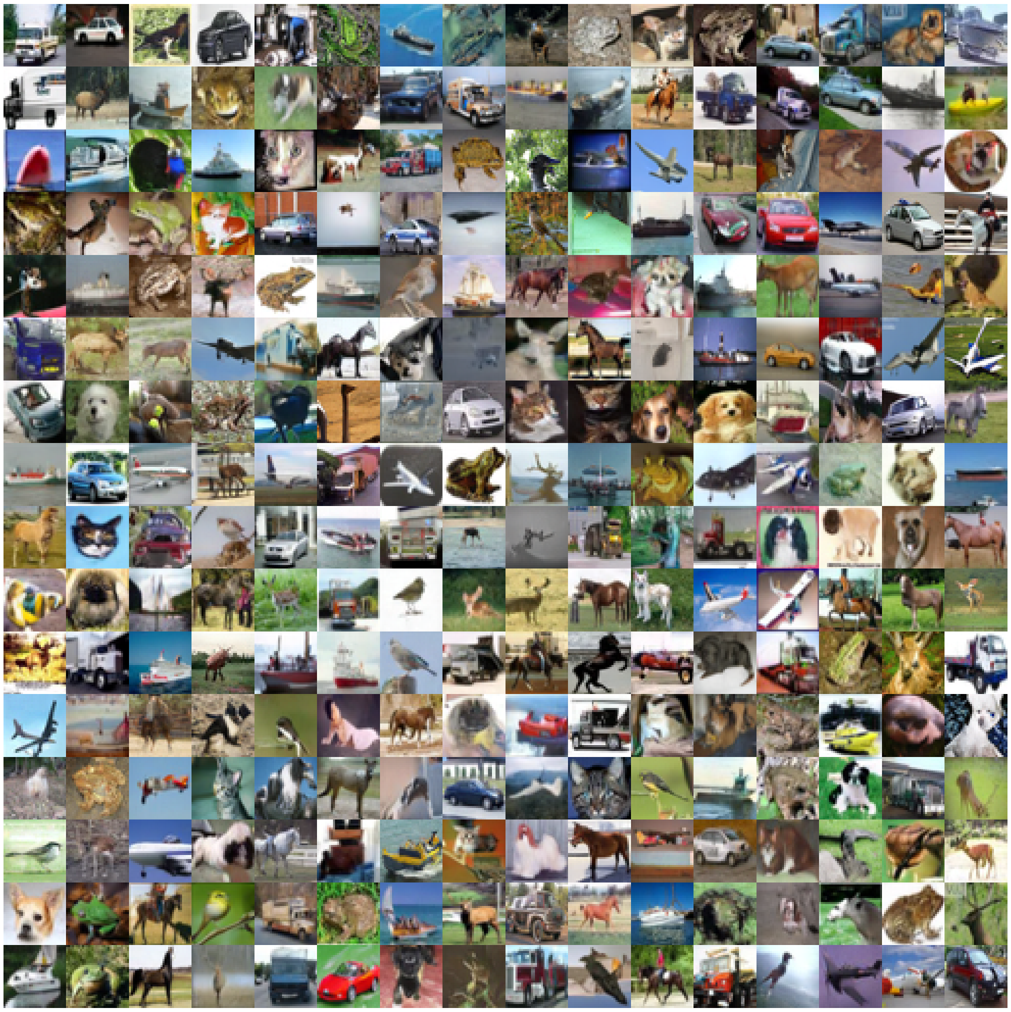


Figure 5: 2-step samples from unconditional SCT trained on CIFAR-10.

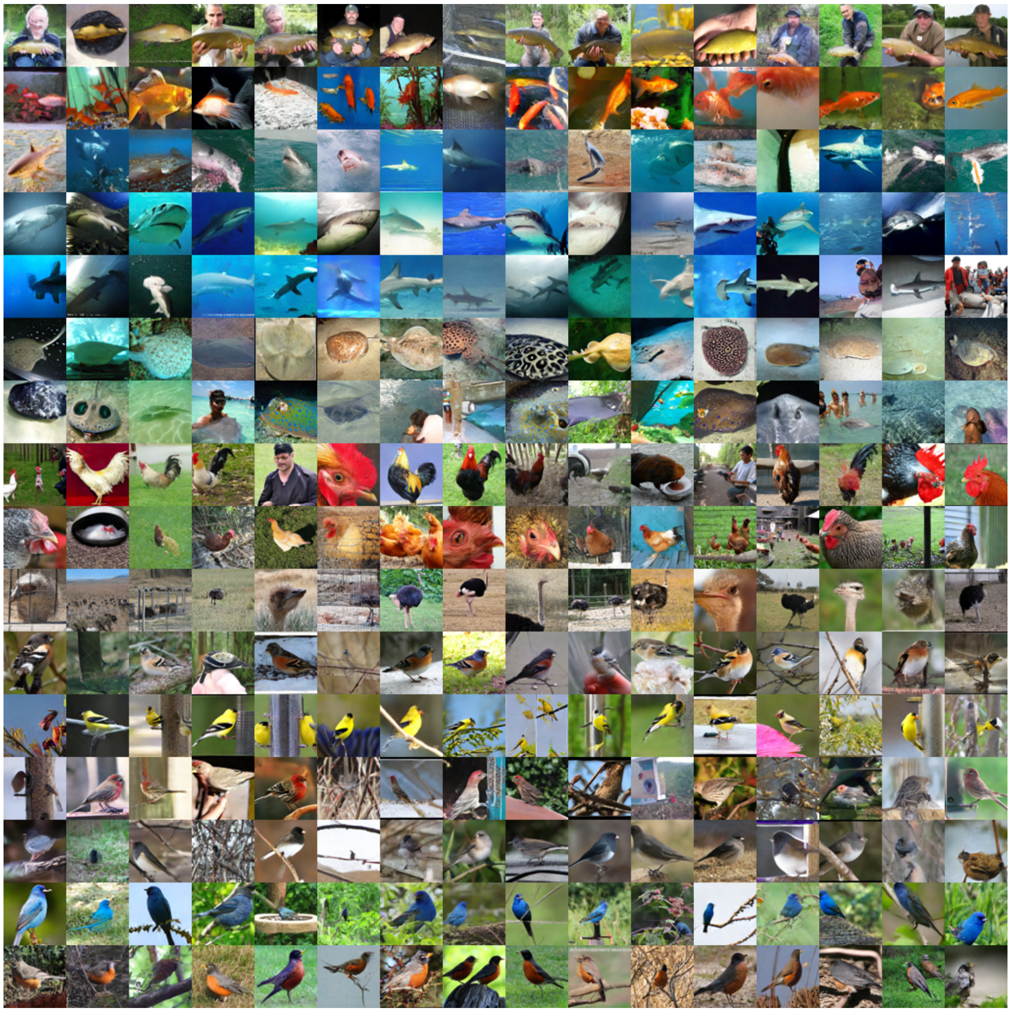


Figure 6: 1-step samples from class-conditional SCT trained on ImageNet-64. Each row corresponds to a different class.

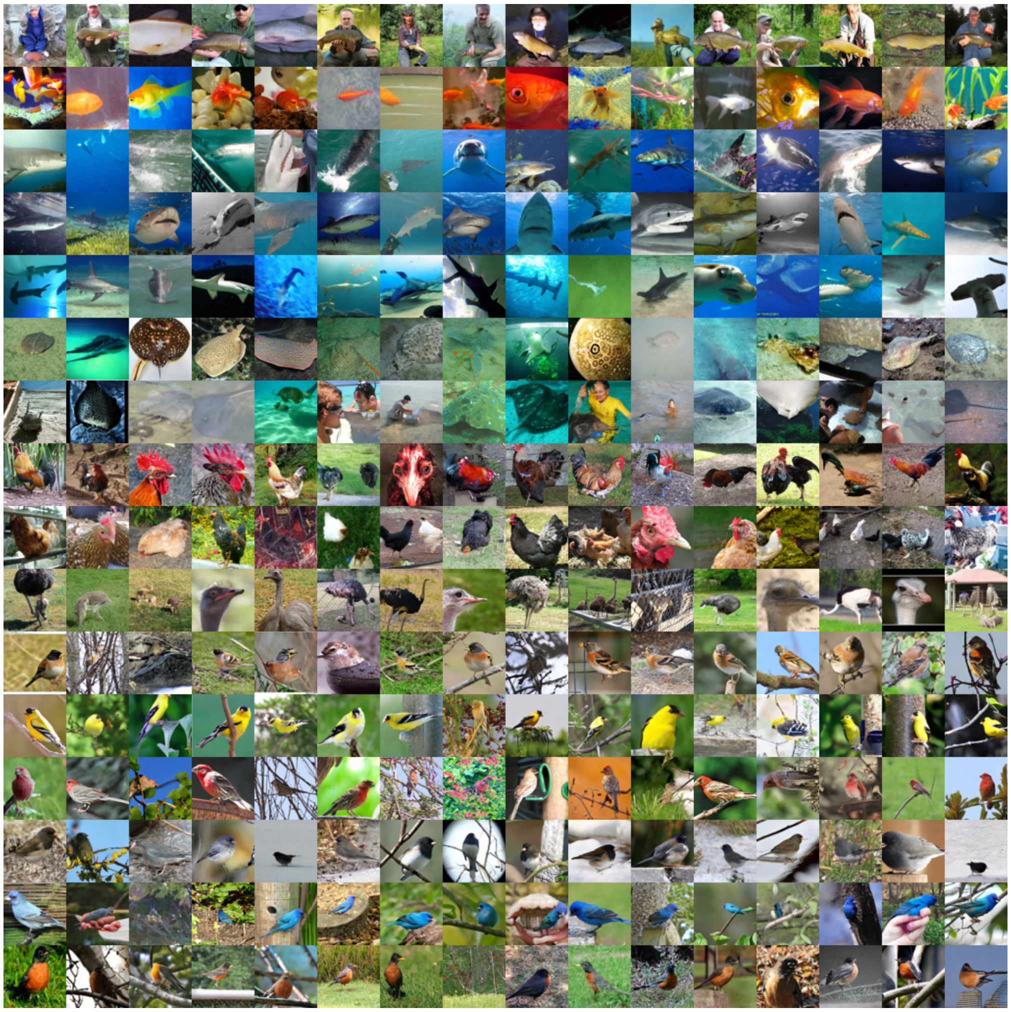


Figure 7: 2-step samples from class-conditional SCT trained on ImageNet-64. Each row corresponds to a different class.