Stabilizing the Training of Consistency Models with Score Guidance

Jeongjun Lee [*1] **Jonggeon Park** [*1] **Jongmin Yoon** [1] **Juho Lee** [1 2]

Abstract

Consistency models exhibit superior sample quality with few steps of sampling, even without relying on pre-trained teacher diffusion models. However, as the number of total discretization steps increases, they suffer from unstable training due to large variance which leads to suboptimal performance. It is known that this can be mitigated by initializing their weights with pre-trained diffusion models, which suggests the potential effectiveness of adopting diffusion models to solve the problem. Inspired by this, we introduce a transformation layer termed *score head*, which is trained in conjunction with consistency model to form a larger diffusion model. Additionally updating consistency model with gradients coming from score head reduces variance during training. We also observe that this joint training scheme aids consistency model to learn common low-level features acquired by diffusion model. The sample quality improves accordingly when measured on CIFAR-10.

1. Introduction

Diffusion Models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have emerged as a de-facto choice for high-fidelity image generation (Dhariwal & Nichol, 2021; Rombach et al., 2022; Karras et al., 2022; 2023). The superior scalability of DMs for high-dimensional data allows their success to extend beyond image synthesis, finding active applications in various domains including point clouds, graphs, texts, and neural network weights (Luo & Hu, 2021; Jo et al., 2022; Huang et al., 2022; Li et al., 2022; Erkoç et al., 2023).

Nonetheless, for real-time applications, the potential of DM is hindered by the sequential nature of its sampling process, as it demands considerable Number of Function Evaluations (NFE). Improving the ODE solvers or introducing new distillation techniques (Dockhorn et al., 2022a; Lu et al., 2022; Zhang & Chen, 2022; Luhman & Luhman, 2021; Salimans & Ho, 2022) have been proposed to remedy the issue.

Among these, Song et al. (2023) proposed Consistency Models (CMs) that map the perturbed data at any noise scale into the original data along the Probability Flow ODE (PF-ODE) trajectory. That is to say, given the PF-ODE

$$\frac{d\mathbf{x}_t}{dt} = -t\boldsymbol{s}(\mathbf{x}_t, t), \quad t \in [0, T]. \tag{1}$$

the CM, denoted $\boldsymbol{f}_\theta$, learns to approximate the following integration

$$\boldsymbol{f}_\theta(\mathbf{x}_t, t) \approx \mathbf{x}_t + \int_t^\epsilon -\tau \boldsymbol{s}(\mathbf{x}_\tau, \tau) d\tau \tag{2}$$

where $\boldsymbol{s}(\mathbf{x}_t, t)$ is the score and $\epsilon > 0$ is a small number to avoid numerical instability. Here, the score is obtained either from a teacher DM in distillation scenarios, or estimated using the unbiased estimator (Efron, 2011)

$$\boldsymbol{s}(\mathbf{x}_t, t) = \frac{\mathbb{E}[\mathbf{x}_0 \,|\, \mathbf{x}_t] - \mathbf{x}_t}{t^2} \tag{3}$$

upon training in isolation.

The second way, namely consistency training, is achieved by minimizing the following objective

$$\mathcal{L}_{\text{CM}}(\theta) = \mathbb{E}\left[\lambda_{\text{CM}}\, d(\boldsymbol{f}_\theta(\mathbf{x}_0 + t_n\mathbf{z}, t_n), \boldsymbol{f}_{\theta^-}(\mathbf{x}_0 + t_{n-1}\mathbf{z}, t_{n-1}))\right] \tag{4}$$

where $\theta^- = \texttt{stopgrad}(\theta)$, $\{t_n\}_{i=1}^N$ are the discretization steps, λ is the weighting function and $d(\mathbf{x}, \mathbf{y})$ is the metric to compare between two vectors $\mathbf{x}$ and $\mathbf{y}$. Here, the training stability depends heavily on the total number of discretization steps N. Setting N to be large, ideally to infinity, is desirable as it introduces lesser bias but this severely disturbs the training due to the increased variance.

Two methods are known to be effective in mitigating the high variance problem. The first is to anneal the model by gradually increasing N as training proceeds, and the second is to initialize the weights of the neural network with

[*]Equal contribution [1]Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology, Seoul, Republic of Korea [2]AITRICS, Seoul, Republic of Korea. Correspondence to: Juho Lee <juholee@kaist.ac.kr>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

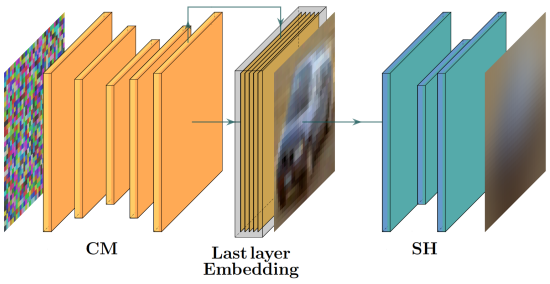

Figure 1. Overall structure of the proposed method. The CM part is trained to be a consistency model as usual, while it is also trained as a part of a larger diffusion model together with the SH. The output and features from CM's U-Net is fed into SH.

that of a pre-trained teacher DM (Song et al., 2023; Song & Dhariwal, 2024). While the former has been successfully adopted in consistency training scenarios, the latter is limited in that it can only be applied in consistency distillation scenarios where teacher models are available.

We aim to supplement this limitation by regularizing the CM to be a part of a larger diffusion model. Specifically, we introduce an additional layer called *score head* at the rear of the CM, and train these two together as a whole to be a diffusion model. Then, the training will be dominated by gradients from DM loss in the early stages of training as the variance of CM loss' gradients is small, resulting in the neural network closer to a DM. Afterwards, the gradients from CM loss will dominate as its variance increases. The intuition is that this will effectively bring similar results as initializing the network with a pre-trained DM.

In this paper, we first introduce a novel aggregation of consistency and diffusion models, while justifying how our modification elucidates a connection between the ground-truth consistency model and the score. Then, we empirically demonstrate that training consistency model with score head improves its performance. The overall structure of our method is outlined in Figure 1.

2. Methods

2.1. Aggregating Score Head with Consistency Model

Differentiating (2) with respect to t yields the following relationship between CM and DM (Song et al., 2023)

$$\frac{\partial \boldsymbol{f}(\mathbf{x}_t, t)}{\partial t} - t \frac{\partial \boldsymbol{f}(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \boldsymbol{s}(\mathbf{x}_t, t) = \mathbf{0} \tag{5}$$

where $\boldsymbol{f}(\mathbf{x}_t, t)$ is the ground-truth CM and $\boldsymbol{s}(\mathbf{x}_t, t)$ is the score. As the ground-truth CM $\boldsymbol{f}(\cdot, t)$ is invertible at $t \in [\epsilon, T]$, we can express the score using the ground-truth CM. However, in practice, as it is not guaranteed that the CM neural network $\boldsymbol{f}_\theta(\cdot, t)$ is invertible, we instead take the pseudo-inverse of the Jacobian to approximate the inverse Jacobian term as

$$\boldsymbol{s}_\theta(\mathbf{x}_t, t) = \frac{1}{t} \left(\frac{\partial \boldsymbol{f}_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right)^\dagger \frac{\partial \boldsymbol{f}_\theta(\mathbf{x}_t, t)}{\partial t} \tag{6}$$

where $A^\dagger$ denotes the pseudo-inverse of the matrix A. Needless to say, computing (6) is highly infeasible since the dimension of the Jacobian of CM would be extremely large. Nevertheless, (6) still suggests that we can obtain the score by transforming CM at a single instance of the input $(\mathbf{x}_t, t)$.

We therefore introduce a transformation layer $\boldsymbol{h}_\phi$ that transforms CM into DM.

$$\boldsymbol{h}_\phi \circ \boldsymbol{f}_\theta(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] \tag{7}$$

In other words, $\boldsymbol{h}_\phi \circ \boldsymbol{f}_\theta$ as a whole is trained to approximate the denoiser while $\boldsymbol{f}_\theta$ alone still learns to be CM on its own. The particular choice of parametrizing the neural network to predict the denoiser was adopted from Karras et al. (2022), as it is known to improve the performance by maintaining the variance of neural network outputs within a narrow scale. An approximation of the score $\boldsymbol{s}_{\theta,\phi}$ can then be obtained using Tweedie's formula (3).

$$\boldsymbol{s}_{\theta,\phi}(\mathbf{x}_t, t) = \frac{\boldsymbol{h}_\phi \circ \boldsymbol{f}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t}{t^2} \tag{8}$$

Along with CM inputs $\mathbf{x}_t$ and t, $\boldsymbol{h}_\phi$ receives features $\mathsf{F}_\theta(\mathbf{x}_t, t)$ produced from the CM neural network as its inputs.

$$\boldsymbol{h}_\phi \circ \boldsymbol{f}_\theta(\mathbf{x}_t, t) = \boldsymbol{h}_\phi(\mathbf{x}_t, t, \mathsf{F}_\theta(\mathbf{x}_t, t)). \tag{9}$$

Specifically, $\mathsf{F}_\theta(\mathbf{x}_t, t)$ include *(i)* `stopgrad` of the CM output $\texttt{stopgrad}(\boldsymbol{f}_\theta(\mathbf{x}_t, t))$, *(ii)* timestep embedding of CM and *(iii)* features from the last ResNet block of CM's U-Net. Such a configuration was largely adopted from Dockhorn et al. (2022b), where they proposed to distill an additional head from the pre-trained diffusion model to approximate higher-order gradients of the score. Utilizing the features of neural networks for other downstream tasks has also been widely adopted in the diffusion model literature (Baranchuk et al., 2021; Luo et al., 2023). We refer to the proposed transformation layer as Score Head (SH).

2.2. Jointly Training CM and SH

CM and SH are jointly trained to minimize the following loss

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{CM}}(\theta) + \mathcal{L}_{\text{SH}}(\theta, \phi) \tag{10}$$

where $\mathcal{L}_{\text{CM}}(\theta)$ is as defined in (4),

$$\mathcal{L}_{\text{SH}}(\theta, \phi) = \mathbb{E}\left[\lambda_{\text{SH}} \left\| \boldsymbol{h}_\phi(\mathbf{x}_t, t, \mathsf{F}_\theta(\mathbf{x}_t, t)) - \mathbf{x}_0 \right\|_2^2 \right] \tag{11}$$

Algorithm 1 Jointly Training CM and SH

Require: Initial CM parameters θ and SH parameters ϕ, learning rates η_1 and η_2, total number of discretization steps $N(\cdot)$, noise schedule distribution $\mathcal{P}$, metric $d(\cdot,\cdot)$, weighting functions λ_{CM} and λ_{SH}.
$k \leftarrow 0$
repeat
Sample $\mathbf{x}_0 \sim p_{\text{data}}$, $\mathbf{z} \sim \mathcal{N}(0, I)$, $n \sim \mathcal{P}[1, N(k)]$
$\mathbf{x}_{t_n} \leftarrow \mathbf{x}_0 + t_n \mathbf{z}$
$\mathbf{x}_{t_{n-1}} \leftarrow \mathbf{x}_0 + t_{n-1}\mathbf{z}$
$\theta^- \leftarrow \texttt{stopgrad}(\theta)$

$\mathcal{L}_{\text{CM}}(\theta) \leftarrow \lambda_{\text{CM}}\, d(\boldsymbol{f}_\theta(\mathbf{x}_{t_n}, t_n), \boldsymbol{f}_{\theta^-}(\mathbf{x}_{t_{n-1}}, t_{n-1}))$

$\hat{\mathbf{x}} \leftarrow \texttt{stopgrad}(\boldsymbol{f}_\theta(\mathbf{x}_{t_n}, t_n))$
$\mathsf{F}_\theta(\mathbf{x}_{t_n}, t_n) \leftarrow \{\hat{\mathbf{x}}, \texttt{timestep embedding}, \texttt{last U-Net block output}\}$

$\mathcal{L}_{\text{SH}}(\theta, \phi) \leftarrow \lambda_{\text{SH}} \|\boldsymbol{h}_\phi(\mathbf{x}_{t_n}, t_n, \mathsf{F}_\theta(\mathbf{x}_{t_n}, t_n)) - \mathbf{x}_0\|_2^2$

$\mathcal{L}(\theta, \phi) \leftarrow \mathcal{L}_{\text{CM}}(\theta) + \mathcal{L}_{\text{SH}}(\theta, \phi)$
$(\theta, \phi) \leftarrow \left(\theta - \eta_1 \frac{\partial \mathcal{L}(\theta,\phi)}{\partial \theta}, \phi - \eta_2 \frac{\partial \mathcal{L}(\theta,\phi)}{\partial \phi}\right)$
$k \leftarrow k + 1$
until θ, ϕ are converged

Table 1. FID scores of various score-based generative models measured on CIFAR-10. The rows marked with * represent our re-implementation.

Method	NFE	FID (↓)	IS (↑)
Diffusion-based			
DDPM (Ho et al., 2020)	1000	3.17	9.46
DDIM (Song et al., 2020)	50	4.67	-
EDM (Karras et al., 2022)	35	2.01	-
Distillation-based			
KD (Luhman & Luhman, 2021)	1	9.36	-
PD (Salimans & Ho, 2022)	1	9.12	-
PD (Salimans & Ho, 2022)	2	4.51	-
CD (Song et al., 2023)	1	3.55	9.48
CD (Song et al., 2023)	2	2.93	9.75
Consistency Models			
CT (Song et al., 2023)	1	8.70	8.49
CT (Song et al., 2023)	2	5.83	8.85
iCT (Song & Dhariwal, 2024)	1	2.83	9.54
iCT (Song & Dhariwal, 2024)	2	2.46	9.80
Ours			
iCT*	1	3.55	9.50
iCT*	2	2.84	9.65
iCT (+ SH) (Ours)	1	**3.37**	**9.53**
iCT (+ SH) (Ours)	2	**2.79**	**9.74**

is the denoising score matching objective and λ_{SH} is a weighting function. Note that minimizing the $\mathcal{L}_{\text{SH}}$ will update not only SH but also CM, as the gradients from SH flow through $\mathsf{F}_\theta(\mathbf{x}, t)$. However, as the gradients directly affect the CM output $\boldsymbol{f}_\theta(\mathbf{x}, t)$ we observe that it is severely damaged occasionally, and thus the `stopgrad` operation is taken before being fed into SH. The overall training scheme is depicted in Algorithm 1.

We conjecture that initializing the weights of CM with a pre-trained DM is effective because both models might share similar low-level features. Given some hints on how to synthesize the denoised images, CMs would more concretely be able to learn the remaining parts. Based on this, we further suppose that the proposed joint training scheme, i.e., training the CM to be a part of a larger DM, benefits CM to more easily learn the common low-level features between CM and DM. As this is achieved by incorporating an additional denoising score matching objective $\mathcal{L}_{\text{SH}}$ on CM, we refer to it as *score guidance*. We observe that the outputs of CM trained in this manner align more closely with ground-truth denoiser, as shown in § 3.

3. Experiments

For the subsequent experiments, we largely employed implementation details from iCT (Song & Dhariwal, 2024) and used our re-implementation of the model upon training CMs, as the official code is unavailable at the time of writing. For SH, we adopted the U-Net architecture of VP-EDM (Karras et al., 2022), but with fewer parameters by reducing the number of ResNet blocks from 4 to 1. Further details are available in Appendix B.

3.1. Increased Training Stability of CM

We measured L_2 norm of the gradients of CM trained with and without score guidance, as shown in the left-hand side of Figure 2. As the total number of discretization steps doubles every 50,000 training steps, the gradient norm also jumps suddenly every 50,000 steps. We thus calculated the variance of measured gradient norms for each interval of 50,000 training steps accordingly.

As the training progresses, the variance of CM trained with score guidance goes lower than that of vanilla iCT, which indicates a more stable training. This further leads to a more improved sample quality, i.e., a lower FID score, demonstrating the superiority of score guidance in reducing the variance in the later stages of training.

3.2. Alignment of CM and SH

To verify that score guidance leads CM to learn common low-level features between CM and DM, we evaluated the cosine similarity between the outputs from jointly trained CM and SH given the same inputs, across different noise

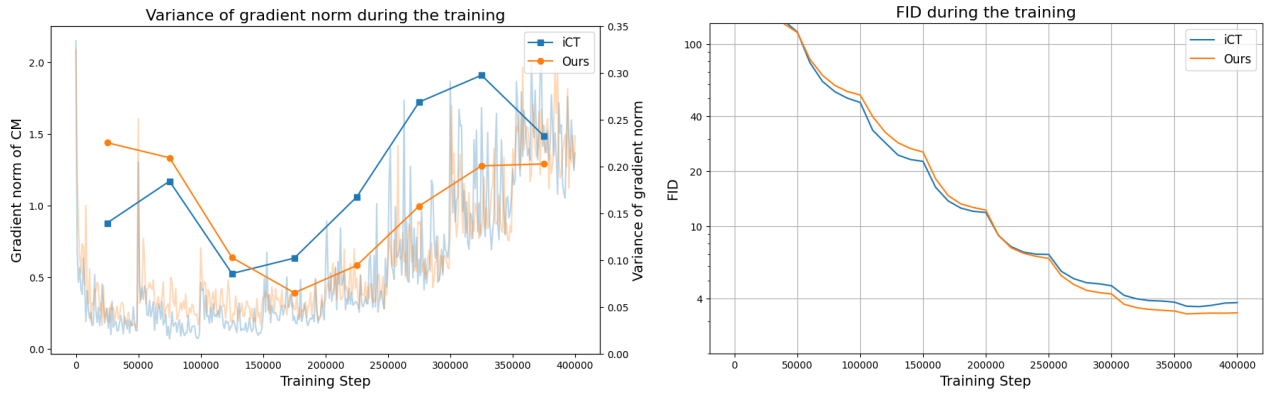

Figure 2. (Left) The gradient norm of CM (dimmed) and its variance (vivid) over the training. The variance is measured for every interval of 50,000 steps. Jointly training SH reduces the variance of the gradient norm in later stages of training. (Right) Evolution of FID scores during training with CIFAR-10. Our joint training scheme excels the baseline after 200,000 steps.

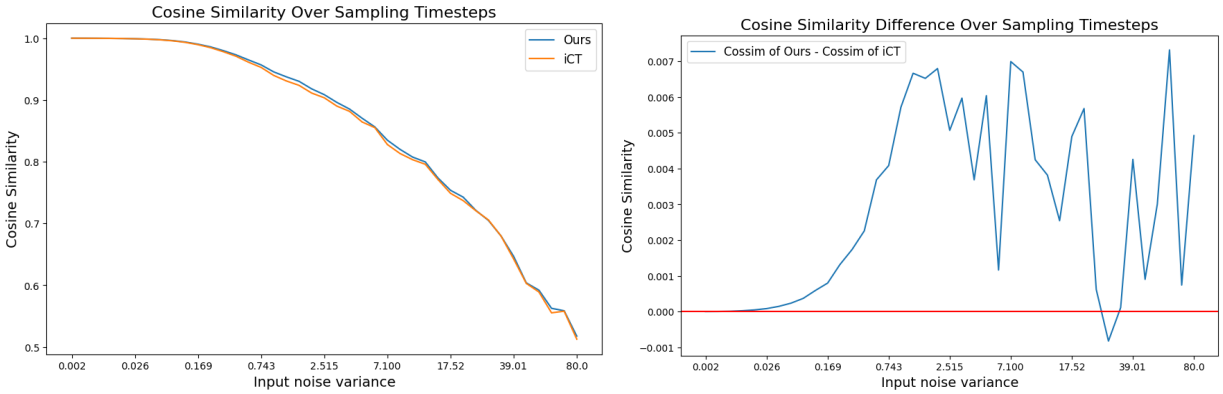

Figure 3. (Left) Cosine similarity between the output of SH and (i) the output of CM jointly trained with SH, (ii) the output of a separately trained iCT. (Right) The difference of the two plots shown on the left. Cosine similarity between the jointly trained CM and SH is higher across almost all noise scales, suggesting that SH indeed guides CM to learn similar low-level features with that of a DM.

scales. Compared to the cosine similarity between the outputs of CM trained separately, CM trained with score guidance show higher correlations with the outputs of SH. This supports our hypothesis that score guidance might assist CM to learn the common low-level features of a DM, thereby leading to improved sample quality.

3.3. Quantitative Results

As shown in Table 1, the proposed method achieves FID scores of 3.37 with one-step generation and 2.79 with two-step generation on the unconditional CIFAR-10 generation task, which outperforms our re-implemented iCT that achieved FID scores of 3.55 and 2.84 for one-step and two-step generation respectively. As we were not able to reproduce iCT, this falls short of the reported FID scores of 2.83 and 2.46. Nonetheless, we still expect that applying our method to the original iCT implementation will further enhance performance.

4. Conclusion

In this paper, we proposed a novel method to stabilize the training of Consistency Model (CM) and improve the sample quality. Specifically, we introduced an additional layer called the Score Head (SH), which takes the CM's output and features as its inputs to approximate the score. In addition to minimizing the CM loss, our method also leverages the gradients of denoising score matching objective coming from SH to train CM. We also demonstrated that SH guides CM to learn common low-level features between CM and DM by observing that the output of CM and SH are highly correlated. Experiment results show that score guidance reduces the variance of CM during training, thereby stabilizing the training process and improving the sample quality when measured on CIFAR-10.

References

Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., and Babenko, A. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2021. 2

Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 1997. 7

Daras, G., Dagan, Y., Dimakis, A., and Daskalakis, C. C. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. 7

Dhariwal, P. and Nichol, A. Q. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021. 1, 7

Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations (ICLR)*, 2022a. 1

Dockhorn, T., Vahdat, A., and Kreis, K. Genie: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022b. 2

Efron, B. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 2011. 1

Erkoç, Z., Ma, F., Shan, Q., Nießner, M., and Dai, A. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015*, 2023. 1

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. 7

Ho, J. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020. 1, 3

Huang, H., Sun, L., Du, B., Fu, Y., and Lv, W. Graphgdp: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, 2022. 1

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of The 39th International Conference on Machine Learning (ICML 2022)*, 2022. 1

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. 1, 2, 3, 7

Karras, T., Hellsten, J., Aittala, M., Aila, T., Lehtinen, J., and Laine, S. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2023. 1

Kim, D., Kim, Y., Kwon, S. J., Kang, W., and Moon, I. Refining generative process with discriminator guidance in score-based diffusion models. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023. 7

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 7

Krizhevsky, A. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. 8

Lai, C., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023a. 7

Lai, C.-H., Takida, Y., Uesaka, T., Murata, N., Mitsufuji, Y., and Ermon, S. On the equivalence of consistency-type models: Consistency models, consistent diffusion models, and fokker-planck regularization. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023b. 7

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. 1

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. 1

Luhman, E. and Luhman, T. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 1, 3

Luo, G., Dunlap, L., Park, D. H., Holynski, A., and Darrell, T. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. 2

Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of The 39th International Conference on Machine Learning (ICML 2022)*, 2022. 7

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 3

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016. 7

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodyanmics. In *Proceedings of The 32nd International Conference on Machine Learning (ICML 2015)*, 2015. 1

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2020. 3

Song, Y. and Dhariwal, P. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 7

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 7

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023. 1, 2, 3

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 7

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations (ICLR)*, 2022. 1

A. Related Works

Links between CM and score The keen relationship between CM and score, *e.g.* (5) which is our primary motivation for the proposed method, can be viewed as a consequence of the consistency property of the score. Daras et al. (2023) demonstrated the consistency property of the score and aimed to enhance the sample quality of diffusion models by incorporating an additional regularization to the denoising score matching objective. Lai et al. (2023b) pointed out that satisfying the consistency property is equivalent to fulfilling score-FPE (Lai et al., 2023a), and that training consistency models can be viewed as enforcing the consistency property along PF-ODE. We note that following the derivation of the PDE in Daras et al. (2023), *i.e.*, the PDE that denoiser should satisfy, for the case of PF-ODE yields (5).

Guidance on score-based model Improving diffusion models with guidance of auxiliary metrics have been widely considered since the emergence of score-based generative models (Song et al., 2021). Inspired by the success of class-conditional Generative Adversarial Network (GAN)s, Dhariwal & Nichol (2021) proposed *classifier guidance* that guides the sampling process towards arbitrary class labels. However, as introducing an additional classifier over noisy data complicates the training procedure, Ho (2022) introduced the *classifier-free guidance* that directly mixes the gradients of conditional and unconditional diffusion models. Kim et al. (2023) improved the performance of the diffusion model with *discriminator guidance* based on the adversarial loss widely covered in the GAN literature. Finally, benefitting from the recent success of large language models, guidance of diffusion model with language models such as CLIP (Nichol et al., 2022) has also been suggested. Guiding CM still remain largely unexplored as they are rather nascent. Yet, Kim et al. (2024) proposed consistency trajectory model (CTM) that naturally includes consistency and diffusion models as its special cases, and showed that incorporating denoising loss upon training CTM indeed leads to improved performance. However, CTM is more complicated than CM as it requires two timesteps as its inputs and relies on adversarial training. To the best of our knowledge, this is the first work that demonstrated the effect of guiding CM without altering its original structure.

B. Experimental Details

Training environment The experiments are conducted on the environment of TPU v3-8 and TPU v4-8. The model architecture and training scheme are implemented with JAX 0.4.20, Flax 0.8.3, and Optax 0.2.2.

Architecture of CM For CIFAR-10, we re-implemented the CM architecture based on iCT (Song & Dhariwal, 2024), conducting experiments with slightly modified of the hyperparameters. Specifically, we use the value of c in the pseudo-Huber loss (Charbonnier et al., 1997) as 0.003 instead of 0.03. See Appendix C for more details.

Architecture of SH For the SH model architecture, we slightly modified the EDM-preconditioned VP architecture implemented in EDM (Karras et al., 2022): we reduced the number of ResNet blocks for each resolution to 1. Additionally, for the SH, we designed the architecture to apply normalization (Wu & He, 2018) to the last layer embedding of CM used in SH training for training stabilization. Subsequently, we concatenated the perturbed data, outputs of CM, and last layer embedding of CM, and utilized the composite of these three tensors for training. With these settings, the number of parameters of SH is 26M which is less than half the number of parameters compared to CM(56M). Given that the size of the score head is smaller than that used in previous EDM studies, the overfitting problem will not be a problem for training SH, so we set the dropout probability for the score head to 0%.

Training For training, we used the RAdam optimizer with a learning rate of 1×10^{-4} for both CM and SH. We used pseudo-Huber loss as the consistency model loss, L_{CM}. The training batch size was set to 1024 and training was carried out for 400k iterations. The paramters of CM and SH are updated simultaneously within each training iteration. For λ_{CM}, total number of discretization steps N and noise schedule $\mathcal{P}$, we followed (Song & Dhariwal, 2024). λ_{SH} was set to 0.3.

Evaluation FID scores (Heusel et al., 2017) and Inception scores (Salimans et al., 2016) were measured with 50,000 samples for both one-step and two-step generation tasks.

C. Additional Experiments

C.1. Ablation Studies

We aimed to find optimal hyperparameters and experimental settings for the proposed method using CIFAR-10 dataset (Krizhevsky, 2009) with a reduced batch size of 512. Experiments were conducted with varying the values of λ_{SH} in the range of $\{1, 0.3, 0.1, 0.05\}$. After determining the λ_{SH}, we determined the c value in the Pseudo-Huber loss

$$\mathcal{L}_{\text{CM}}(\theta) = \lambda_{\text{CM}} \left(\left(\left(\boldsymbol{f}_{\theta}(\mathbf{x}_{t_n}, t_n) - \boldsymbol{f}_{\theta^-}(\mathbf{x}_{t_{n-1}}, t_{n-1}) \right)^{\otimes 2} + c^2 \right)^{\otimes \frac{1}{2}} - c \right) \tag{12}$$

in the range of $\{0.03, 0.01, 0.003, 0.0003\}$. $\otimes p$ denotes the element-wise power to p.

Experiment results are illustrated in Figure 4, which show that $\lambda_{\text{SH}} = 0.3$ and $c = 0.003$ are the most optimal choice.

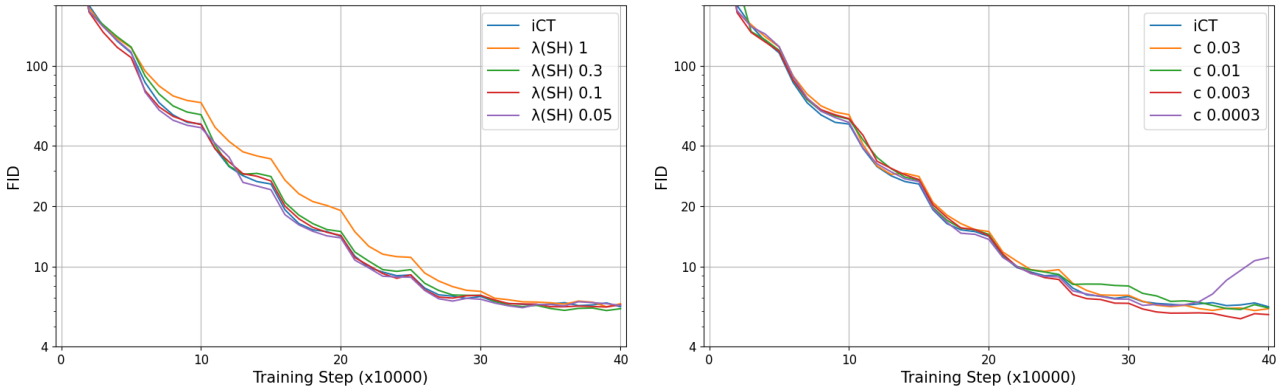

Figure 4. Ablation studies on hyperparameters. (Left) Sample quality over varying λ_{SH}, (Right) Sample quality over varying c. FID scores were calculated using 10,000 samples.

D. Additional Samples

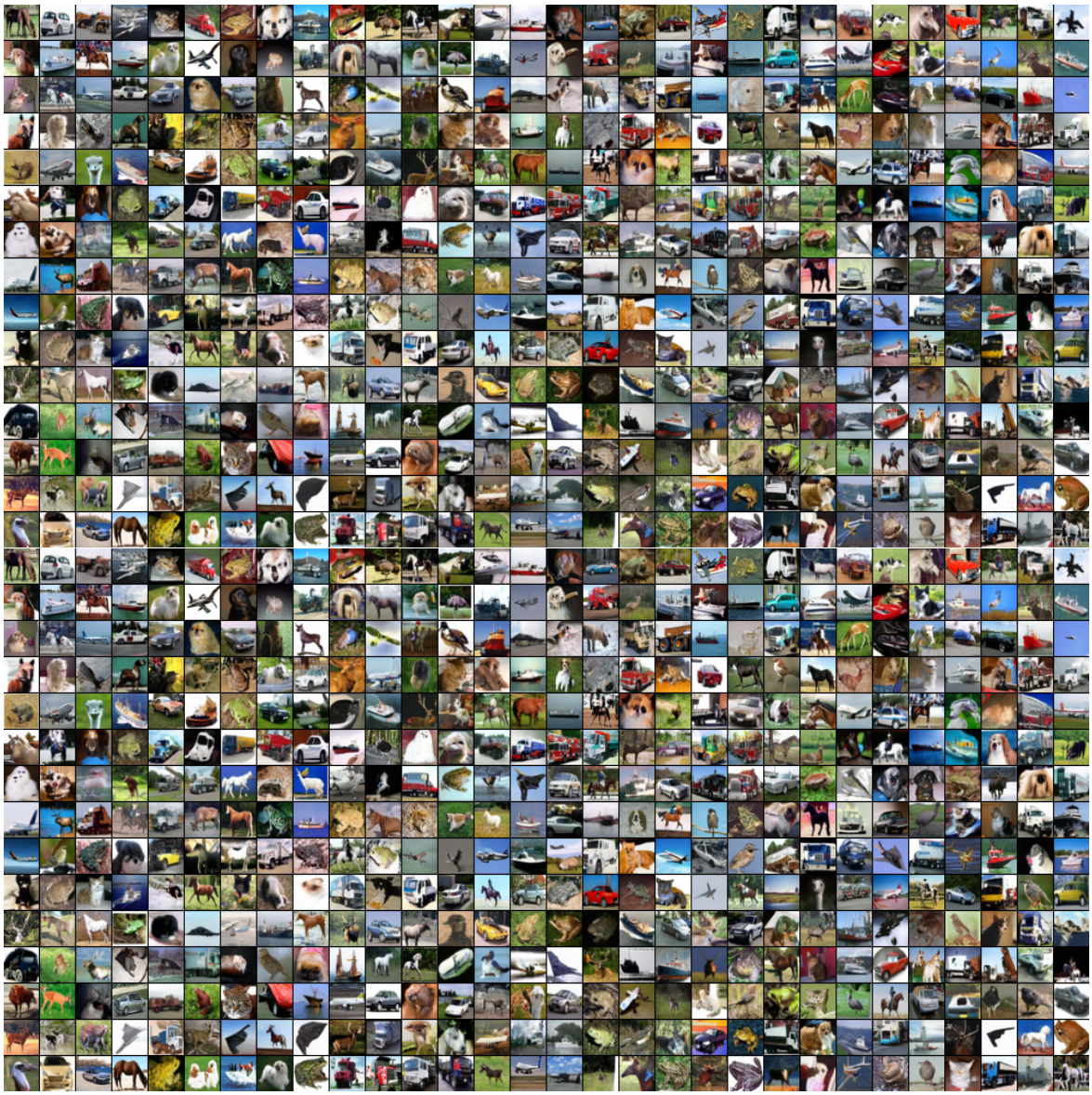

Figure 5. Uncurated generated images on CIFAR-10. (Top) One-step generation. (Bottom) Two-step generation.