Understanding Generalization in Diffusion Models via Probability Flow Distance

Huijie Zhang, Zijian Huang, Siyi Chen
Jinfan Zhou, Zekai Zhang, Peng Wang, Qing Qu
{HUIJIEZH, NAMELESS, SIYICH, ZJF, ZZEKAI, PENGWA, QINGQU}@UMICH.EDU
University of Michigan, Ann Arbor, United States

Abstract

Diffusion models have emerged as a powerful class of generative models, capable of producing high-quality samples that generalize beyond the training data. However, evaluating this generalization remains challenging: theoretical metrics are often impractical for high-dimensional data, while no practical metrics rigorously measure generalization. In this work, we bridge this gap by introducing probability flow distance (PFD), a theoretically grounded and computationally efficient metric to measure distributional generalization. Specifically, PFD quantifies the distance between distributions by comparing their noise-to-data mappings induced by the probability flow ODE. Moreover, by using PFD under a teacher-student evaluation protocol, we empirically uncover several key generalization behaviors in diffusion models, including: (1) scaling behavior from memorization to generalization, (2) early learning and double descent training dynamics, and (3) bias-variance decomposition. Beyond these insights, our work lays a foundation for future empirical and theoretical studies on generalization in diffusion models.

1. Introduction

In recent years, diffusion models and their variants have revolutionized generative AI, achieving state-of-the-art performance in tasks ranging from image synthesis [21, 59] to molecular design [1, 31]. These models, including score-based generative models [66] and flow matching techniques [47, 50], learn the underlying data distribution through forward and reverse processes that gradually inject and remove noise. Their success raises a fundamental question: how can we rigorously evaluate the generalization ability of these models? A good evaluation framework is essential not only for deepening our understanding of the underlying mechanisms of generative modeling but also for providing principled guidance in designing more effective architectures, training strategies, and benchmarking methods.

However, existing metrics for evaluating the generalizability of diffusion models face significant limitations. Empirically, common metrics like Fréchet inception distance (FID) [28], Inception Score (IS) [64] focus on generation quality, but they cannot distinguish between memorization and generalization, as both can yield high-quality outputs. While metrics such as Kullback-Leibler (KL) divergence [16, 44, 54], total variation (TV) [14, 42, 43, 71], and Wasserstein distance [11, 12, 23, 24] are theoretically appealing, they are often computationally expensive and thus impractical for diffusion models. Furthermore, since the true data distribution is typically unknown, it makes such comparisons inherently challenging. In summary, existing metrics are neither accurate nor efficient for evaluating diffusion models in practice, highlighting the need for a generalization metric that is both theoretically grounded and practically tractable.

© H.Z.Z.H.S.C..J.Z.Z.Z.P.W.Q. Qu.

Our Contribution. In this work, we introduce a systematic framework for evaluating the generalizability of diffusion models through a novel metric, the probability flow distance (PFD). This metric quantifies distributional differences by leveraging the backward probability flow ODE (PF-ODE) [66], which is widely used in the sampling process of diffusion models. Unlike practical metrics such as FID, PFD provides a theoretically grounded measure of distance between distributions, offering a more reliable assessment of generalization. Compared to theoretical metrics like the Wasserstein distance, PFD is computationally efficient by leveraging the benign properties of PF-ODE. Moreover, under a distillation-based setting, we use this metric to study generalization error by comparing the PFD between the student and teacher models. Our analysis reveals several intriguing generalization phenomena that offer new insights into the learning behavior of diffusion models, including: (1) Scaling behavior from memorization to generalization. (2) Early learning and double descent of generalization in learning dynamics. (3) Bias and variance trade-off of the generalization error. Due to space limitations, we defer this to Appendix B.

2. Measuring Distribution Distance via Probability Flow Distance

In this section, we propose a new metric called probability flow distance (PFD), which is designed to quantify the distance between two arbitrary probability distributions. The design of PFD is motivated by the PF-ODE, which we first review in Section 2.1. We then formally define PFD in Section 2.2 and present its empirical estimation with theoretical guarantees in Section 2.3.

2.1. A Mapping from Noise to Target Distribution Spaces Induced by PF-ODE

In general, PF-ODE is a class of ordinary differential equations (ODE) that aim to reverse a forward process, where Gaussian noise is progressively added to samples drawn from an underlying distribution, denoted as $p_{\texttt{data}}$ [1]. The forward process and the PF-ODE can be described as follows:

- *Forward process.* Given a sample $\boldsymbol{x}_0 \overset{i.i.d.}{\sim} p_{\texttt{data}}(\boldsymbol{x})$, the forward process progressively corrupts it by adding Gaussian noise. This process can be characterized by the stochastic differential equation (SDE) $\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t$, where $t \in [0, T]$ is the time index, $\{\boldsymbol{w}_t\}_{t\in[0,T]}$ is a standard Wiener process, and $f(t), g(t) : \mathbb{R}_+ \to \mathbb{R}$ are drift and diffusion function functions that control the noise schedule. In this work, we adopt the noise schedule proposed by elucidated diffusion models (EDM) [36], where $f(t) = 0$ and $g(t) = \sqrt{2t}$. Substituting this into the SDE and integrating both sides, we obtain
$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t \sqrt{2\tau}\mathrm{d}\boldsymbol{w}_\tau. \tag{1}$$
For ease of exposition, we use $p_t(\boldsymbol{x}_t)$ to denote the distribution of the noisy image $\boldsymbol{x}_t$ for each $t \in [0, T]$. In particular, it is worth noting that $p_0(\boldsymbol{x}) = p_{\texttt{data}}(\boldsymbol{x})$ and $p_T(\boldsymbol{x}) \to \mathcal{N}(\mathbf{0}, T^2\boldsymbol{I}_n)$ as $T \to +\infty$.

- *Probability flow ODE.* According to [66], the PF-ODE can transform a noise sample $\boldsymbol{x}_T$ back into a clean data sample $\boldsymbol{x}_0$. Specifically, under EDM noise scheduler, the PF-ODE admits the following form:
$$\mathrm{d}\boldsymbol{x}_t = -t\nabla \log p_t(\boldsymbol{x}_t)\mathrm{d}t, \tag{2}$$

1. This paper primarily focuses on image distribution.

where $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$ (or simply $\nabla \log p_t(\boldsymbol{x}_t)$) denotes the *score function* of the distribution $p_t(\boldsymbol{x}_t)$ at time $t \in [0, T]$. According to [66], the backward PF-ODE (2) and the forward SDE (1) have the same distribution at each timestep t. In practice, since the score function $\log p_t(\boldsymbol{x}_t)$ is unknown, in diffusion models we approximate it using a neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ and employ a numerical solver to generate samples from Equation (2). Additional details are provided in Appendix A.3.

Benign properties of PF-ODE. The backward PF-ODE introduces a mapping $\boldsymbol{\Phi}_{p_{\texttt{data}}}$ from $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$. By taking the integral on both sides of (2) from T to 0, the mapping $\boldsymbol{\Phi}_{p_{\texttt{data}}}$ can be defined as:

$$\boldsymbol{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T) \coloneqq \boldsymbol{x}_T - \int_T^0 t \nabla \log p_t(\boldsymbol{x}_t) \mathrm{d}t. \tag{3}$$

Previous work [66] demonstrates that $\boldsymbol{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T) \sim p_{\texttt{data}}(\boldsymbol{x})$ when $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, T^2 \boldsymbol{I}_n)$ as $T \to +\infty$. This implies that when the underlying distribution $p_{\texttt{data}}$ is known, the score function $\nabla \log p_t(\boldsymbol{x}_t)$ becomes explicitly available, and the backward PF-ODE induces a deterministic mapping from the Gaussian distribution to $p_{\texttt{data}}$.

2.2. Definition of Probability Flow Distance

Definition 1 (Probability flow distance (PFD)) *For any two given distributions p and q of the same dimension, we define their distribution distance as*

$$\texttt{PFD}\left(p, q\right) \coloneqq \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2 \boldsymbol{I})}\left[\left\|\boldsymbol{\Psi} \circ \boldsymbol{\Phi}_p\left(\boldsymbol{x}_T\right) - \boldsymbol{\Psi} \circ \boldsymbol{\Phi}_q\left(\boldsymbol{x}_T\right)\right\|_2^2\right]\right)^{1/2}. \tag{4}$$

Here, $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_q$ denote the mappings between the noise and image spaces for distributions p and q, respectively, as defined in (3), and $\boldsymbol{\Psi}(\cdot)$ represents an image descriptor.

Intuitively, PFD measures the distance between two distributions p and q by comparing their respective noise-to-image mappings $\boldsymbol{\Phi}_p(\cdot)$ and $\boldsymbol{\Phi}_q(\cdot)$ starting from the same Gaussian noise input $\boldsymbol{x}_T$.

Moreover, the comparison is conducted in a transformed feature space defined by an image descriptor $\boldsymbol{\Psi}(\cdot)$, which is typically implemented using a pre-trained neural network to effectively capture perceptual differences.

Under Definition 1, we show that PFD satisfies the axioms of a metric (Definition 2.15 in [61]).

Theorem 1 *For any two distributions p and q, the* PFD *satisfies the following properties:*

- *(Positivity)* $\texttt{PFD}(p, q) > 0$ *for any* $p \neq q$.
- *(Identity Property)* $\texttt{PFD}(p, q) = 0$ *if and only if* $p = q$.
- *(Symmetry)* $\texttt{PFD}(p, q) = \texttt{PFD}(q, p)$.
- *(Triangle Inequality)* $\texttt{PFD}(p, q) \leq \texttt{PFD}(p, p') + \texttt{PFD}(p', q)$ *for all* p'.

We defer the proof to Appendix C. Note that Theorem 1 establishes the theoretical validity of PFD as a metric for measuring distance between any two probability distributions.

2.3. Empirical Estimation of PFD

In practice, the expectation in (4) is intractable due to the complexity of the underlying distributions. Thus, we approximate the PFD using finite samples:

$$\hat{\mathtt{PFD}}(p,q) = \left(\frac{1}{M}\sum_{i=1}^{M}\left\|\boldsymbol{\Phi}_p\left(\boldsymbol{x}_T^{(i)}\right) - \boldsymbol{\Phi}_q\left(\boldsymbol{x}_T^{(i)}\right)\right\|_2^2\right)^{1/2}. \tag{5}$$

Here, $\hat{\mathtt{PFD}}(p,q)$ is the empirical version of $\mathtt{PFD}(p,q)$ computed over M independent samples $\{\boldsymbol{x}_T^{(i)}\}_{i=1}^{M} \overset{i.i.d.}{\sim} \mathcal{N}(0, T^2\boldsymbol{I}_n)$ with $T \to \infty$.

Specifically, our finite-sample approximation relies on two key assumptions: (i) the score functions are smooth at all timesteps, and (ii) the score functions of two distributions remain uniformly close within a bounded region of the input space, which can be described as follows.

Assumption 1 *Let p and q be two distributions with the same dimension, where we assume:*

(i) There exists a constant $L > 0$ such that for all $\boldsymbol{x}_1, \boldsymbol{x}_2$ and $t \in [0, T]$, it holds that

$$\|\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_1) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_2)\|_2 \le L\,\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2\,, \tag{6}$$

and similarly for q_t.

(ii) For all $t \in [0, T]$, there exists a constant $\epsilon > 0$ such that

$$\|\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x})\|_2 \le \epsilon. \tag{7}$$

The Lipschitz continuity of the score function is a common assumption widely adopted in the theoretical analysis of score functions in diffusion models [10, 12, 13, 15, 41, 76]. More recently, this property has been rigorously established under the assumption that the data distribution is a mixture of Gaussians [46]. The uniform closeness assumption holds when p, q follow Assumption 1 (i) and have support on a compact domain, which is often the case for image distributions. Under Assumption 1, the concentration of the empirical estimate $\hat{\mathtt{PFD}}(p,q)$ to $\mathtt{PFD}(p,q)$ can be characterized as follows.

Theorem 2 *Suppose we are given two distributions p and q that satisfy the L-Lipschitz condition and are uniformly close in Assumption 1. Let $\hat{\mathtt{PFD}}(p,q)$ denote the empirical estimate of $\mathtt{PFD}(p,q)$, computed as the average over M independent samples, as introduced in (5). Then, for any $\gamma > 0$, the empirical estimate satisfies the following bound:*

$$\left|\hat{\mathtt{PFD}}(p,q) - \mathtt{PFD}(p,q)\right| \le \gamma, \quad \textit{whenever } M \ge \frac{\kappa^4(L,\epsilon)}{2\gamma^4}\log\frac{2}{\eta}, \tag{8}$$

with probability at least $1-\eta$. Here, $\kappa(L,\epsilon) := \exp\left(\frac{LT_\xi^2}{2}\right)\xi + \frac{\epsilon}{L}\left(\exp\left(\frac{LT_\xi^2}{2}\right) - 1\right)$ is a constant, with a numerical constant $\xi > 0$ and a finite timestep T_ξ depending only on ξ.

We defer the proof to Appendix C. Given the score functions of both distributions are smooth and uniformly close, our result in Theorem 2 guarantees that $\mathtt{PFD}(p,q)$ can be approximated to arbitrary precision by its empirical estimate $\hat{\mathtt{PFD}}(p,q)$ with high probability, givem a finite number of samples.

3. Quantifying Generalization Error of Diffusion Models

In this section, we leverage the PFD metric in Section 2 to rigorously define and evaluate the generalization error of diffusion models. Specifically, this metric enables us to distinguish between memorization and generalization behaviors for diffusion models, as well as analyze the transition from memorization to generation (MtoG).

Definition 2 (Generalization and Memorization Errors) *Consider a diffusion model $\boldsymbol{s}_{\boldsymbol{\theta}}$ trained on a finite dataset $\mathcal{D} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{N}$, where each sample $\boldsymbol{y}^{(i)}$ is drawn i.i.d. from the underlying distribution $p_{\texttt{data}}(\boldsymbol{x})$. Denote the learned distribution induced by a diffusion model $\boldsymbol{s}_{\boldsymbol{\theta}}$ as $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Using the* PFD *metric, we can formally define the generalization and memorization errors as follows:*

$$\mathcal{E}_{\texttt{gen}}(\boldsymbol{\theta}) := \texttt{PFD}\left(p_{\boldsymbol{\theta}}, p_{\texttt{data}}\right), \quad \mathcal{E}_{\texttt{mem}}(\boldsymbol{\theta}) := \texttt{PFD}\left(p_{\boldsymbol{\theta}}, p_{\texttt{emp}}\right), \tag{9}$$

where the empirical distribution is given by $p_{\texttt{emp}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{y}^{(i)})$, with $\delta(\cdot)$ denoting the Dirac delta function.

Here, given access to $p_{\texttt{emp}}(\boldsymbol{x})$, the memorization error $\mathcal{E}_{\texttt{mem}}(\boldsymbol{\theta})$ can be exactly computed (see Appendix E). We further show that $\mathcal{E}_{\texttt{mem}}(\boldsymbol{\theta})$ coincides with metrics introduced in [73, 75]. However, since the underlying distribution $p_{\texttt{data}}(\boldsymbol{x})$ is typically unknown in practice, we introduce a teacher–student evaluation protocol to analyze the generalization error of diffusion models.

Evaluation protocol of generalization. To study the generalization behavior of diffusion models (see Appendix B), we adopt a teacher–student framework. We treat a large-scale pretrained diffusion model $\boldsymbol{s}_{\boldsymbol{\theta}_{\texttt{t}}}(\boldsymbol{x}_t)$ with parameters $\boldsymbol{\theta}_{\texttt{t}}$ as the teacher, inducing a distribution $p_{\boldsymbol{\theta}_{\texttt{t}}}$, which we take as a proxy for the true data distribution, i.e., $p_{\texttt{data}} = p_{\boldsymbol{\theta}_{\texttt{t}}}$. We then train a student model $\boldsymbol{s}_{\boldsymbol{\theta}}$ using samples drawn from $p_{\boldsymbol{\theta}_{\texttt{t}}}$, and evaluate its generalization by comparing $p_{\boldsymbol{\theta}}$ to $p_{\boldsymbol{\theta}_{\texttt{t}}}$ using the generalization errors defined in Definition 2.

In our experiments for the rest of the paper, both teacher and student models adopt the U-Net architecture [60]. The teacher model $\boldsymbol{s}_{\boldsymbol{\theta}_{\texttt{t}}}$ is trained on the CIFAR-10 dataset [39] with a fixed model architecture (UNet-10 introduced in Appendix D.1). The student model $\boldsymbol{s}_{\boldsymbol{\theta}}$ is trained on samples generated by the teacher, with the number of training samples varying from $N = 2^6$ to $N = 2^{16}$, using the same training hyperparameters but different model sizes. For evaluating the generalization error in (9), we compute the PFD between the teacher and student models using $M = 10^4$ samples drawn from shared initial noise, as defined in (5). Similar for the memorization error, we compute the PFD between the student model and the empirical distribution of the training data. Additional details for the evaluation protocol and ablation studies with different teacher models are provided in Appendix D.2 and Appendix F.4, respectively. Due to the space limitation, we defer the generalization behaviors in diffusion models experiment in Appendix B.

4. Conclusion

In this work, we introduced Probability Flow Distance, a theoretically grounded and computationally tractable metric for evaluating the generalization ability of diffusion models. Using a teacher–student evaluation protocol, we empirically reveal several key generalization behaviors in learning diffusion models, including: (1) the scaling transition from memorization to generalization, (2) early learning and double descent training dynamics, and (3) a bias–variance trade-off of generalization error.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[2] Sumukh K Aithal, Pratyush Maini, Zachary Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. In *Advances in Neural Information Processing Systems*, volume 37, pages 134614–134644, 2024.

[3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[4] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[5] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

[6] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[7] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

[8] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[9] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

[10] Adam Block, Youssef Mroueh, and Alexander Rakhlin. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

[11] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=MhK5aXo3gB. Expert Certification.

[12] Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763. PMLR, 2023.

[13] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[14] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ODE is provably fast. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KD6MFeWSAd.

[15] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

[16] Sitan Chen, Giannis Daras, and Alex Dimakis. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. In *International Conference on Machine Learning*, pages 4462–4484. PMLR, 2023.

[17] Zhengdao Chen. On the interpolation effect of score smoothing. *arXiv preprint arXiv:2502.19499*, 2025.

[18] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6070–6079, 2020.

[19] Earl A Coddington and Norman Levinson. *Theory of ordinary differential equations*. McGraw-Hill New York, 1955.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[21] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[22] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025. URL https://openreview.net/forum?id=C8Yyg9wy0s.

[23] Xuefeng Gao and Lingjiong Zhu. Convergence analysis for general probability flow ODEs of diffusion models in wasserstein distances. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=EkO8rb3liX.

[24] Xuefeng Gao, Hoang M. Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *Journal of Machine Learning Research*, 26 (43):1–54, 2025. URL http://jmlr.org/papers/v26/24-0902.html.

[25] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

[26] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

[27] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *Empirical Methods in Natural Language Processing*, 2021.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[30] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The Collected Works of Wassily Hoeffding*, pages 409–426, 1994.

[31] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

[32] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315, 2024.

[33] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0.

[34] Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

[35] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[36] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577, 2022.

[37] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. <https://github.com/NVlabs/edm/tree/main>, 2022.

[38] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *International Conference on Machine Learning*, volume 96, pages 275–283. Citeseer, 1996.

[39] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[40] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[41] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. In *Advances in Neural Information Processing Systems*, volume 35, pages 22870–22882, 2022.

[42] Gen Li, Yu Huang, Timofey Efimov, Yuting Wei, Yuejie Chi, and Yuxin Chen. Accelerating convergence of score-based diffusion models, provably. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[43] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards non-asymptotic convergence for diffusion-based generative models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4VGEeER6W9.

[44] Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pages 2097–2127, 2023.

[45] Xiang Li, Yixiang Dai, and Qing Qu. Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Sk2duBGvrK.

[46] Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Unraveling the smoothness properties of diffusion models: A gaussian mixture perspective. *arXiv preprint arXiv:2405.16418*, 2024.

[47] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[48] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, volume 33, pages 20331–20342, 2020.

[49] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pages 14153–14172. PMLR, 2022.

[50] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[51] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[52] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

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

[54] Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DesYwmUG00.

[55] Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. *arXiv preprint arXiv:2411.19339*, 2024.

[56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

[57] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022.

[58] 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 *International Conference on Machine Learning*, pages 8748–8763. PmLR, 2021.

[59] 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.

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[61] Walter Rudin. Principles of mathematical analysis. 2021.

[62] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[63] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

[64] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[65] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[66] 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.

[67] George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[68] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[69] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[70] Binxu Wang and John J Vastola. The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications. *arXiv preprint arXiv:2412.09726*, 2024.

[71] Ruofeng Yang, Zhijie Wang, Bo Jiang, and Shuai Li. Leveraging drift to improve sample complexity of variance exploding diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pages 107662–107702, 2024.

[72] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pages 10767–10777. PMLR, 2020.

[73] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

[74] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

[75] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60558–60590. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhang24cn.html.

[76] Zhenyu Zhu, Francesco Locatello, and Volkan Cevher. Sample complexity bounds for score-matching: Causal discovery and generative modeling. In *Advances in Neural Information Processing Systems*, volume 36, pages 3325–3337, 2023.

Appendix A. Related Works

In this section, we briefly review related work on generalization metrics for diffusion models, discuss diffusion model generalizability, and cover the fundamentals of training diffusion models.

A.1. Generalization Metrics for Diffusion Models

Generalization metrics quantify the distance between the learned distribution and the underlying data distribution in diffusion models. To measure this distributional gap, theoretical works commonly employ metrics such as Kullback-Leibler (KL) divergence [16, 44, 54], total variation (TV) [14, 42, 43, 71], and Wasserstein distance [11, 12, 23, 24]. However, these metrics are practically inefficient for diffusion models. Practical metrics focus on various perspective, including negative log-likelihood (`NLL`) [66], image generation quality: Fréchet inception distance (`FID`) [28], inception score (`IS`) [64], $\texttt{FD}_{\texttt{dinov2}}$ [67], maximum mean discrepancy (`MMD`) [9], CLIP maximum mean discrepancy (CMMD) [32]; alignment: `CLIPscore` [27], and `precision, recall` [40, 62]. However, these practical metrics are not explicitly designed to evaluate the generalizability of diffusion models. Thus, there is a need for a generalization metric that are both theoretical grounded and practically efficient for diffusion models. To address this gap, we propose `PFD`, a novel generalization metric that is theoretically proven to be a valid distributional distance and can be efficiently approximated by its empirical version using a polynomial number of samples. In practice, `PFD` requires fewer samples for estimation and is the only existing metric that explicitly quantifies generalization in diffusion models.

A.2. Diffusion Model Generalizability

Recent works have shown that diffusion models transition from memorization to generalization as the number of training samples increases [73, 75]. With sufficient data, models trained with different architectures, loss functions, and even disjoint datasets can reproduce each other's outputs, indicating a strong convergence toward the underlying data distribution [33, 75]. To explain this strong generalization, [33] attribute it to the emergence of a geometric-adaptive harmonic basis, while others argue that generalization arises from interpolation across the data manifold [2, 17]. Theoretical insights by [44] provide generalization bounds using KL-divergence under simplified models. More recent efforts focus on characterizing the learned noise-to-image mapping for generalized diffusion models, either through Gaussian parameterizations [45, 70] or patch-wise optimal score functions [34, 55]. However, despite these theoretical analyses and qualitative insights, prior work lacks a quantitative framework for measuring generalizability. In this paper, we propose `PFD`, a metric that enables such quantitative evaluation. Using this measure, we uncover further insights into the generalization behavior of diffusion models, as discussed in Appendix B.

A.3. Training diffusion models

To enable sampling via the PF-ODE (2), we train a neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ to approximate the score function $\nabla \log p_t(\boldsymbol{x}_t)$ using denoising score matching loss [66]:

$$\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \int_0^T \lambda_t \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, T^2 \boldsymbol{I}_n)} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} + t\boldsymbol{\epsilon}, t) + \boldsymbol{\epsilon}/t \right\|_2^2 \right] \mathrm{d}t, \tag{10}$$

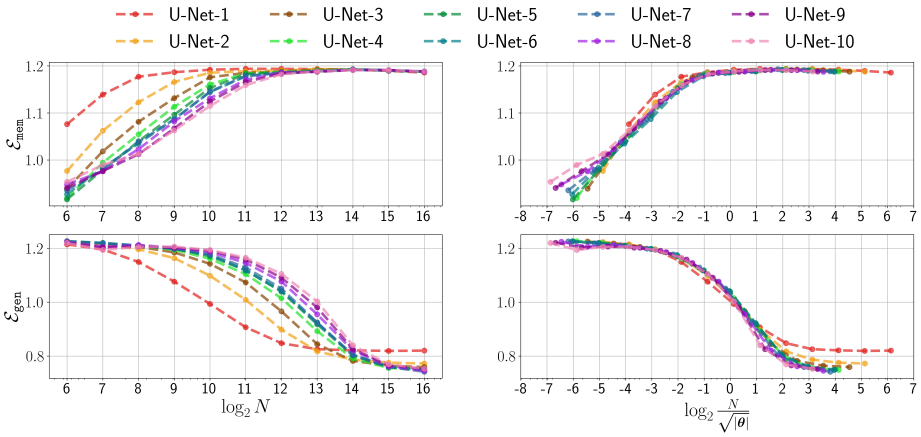

Figure 1: **Scaling behavior in the MtoG transition.** $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ plotted against $\log_2(N)$ for a range of U-Net architectures (U-Net-1 to U-Net-10). Right: the same metrics plotted against $\log_2(N/\sqrt{|\boldsymbol{\theta}|})$, where $|\boldsymbol{\theta}|$ is the number of model parameters.

λ_t denotes a scalar weight for the loss at t. Given the learned score function, the corresponding noise-to-image mapping is:

$$\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T) = \boldsymbol{x}_T - \int_T^0 t\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\mathrm{d}t. \tag{11}$$

Although alternative training objectives exist, such as predicting noise $\boldsymbol{x}_T$ [29], clean image $\boldsymbol{x}_0$ [36], rectified flow $\boldsymbol{x}_T - \boldsymbol{x}_0$ [50] or other linear combinations of $\boldsymbol{x}_0$ and $\boldsymbol{x}_T$ [63], prior works [22, 51] have shown that it is still possible to recover an approximate score function $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ from these methods.

Appendix B. Measuring Key Generalization Behaviors in Diffusion Models

Based on the evaluation protocol in Section 3, this section reveals several key generalization behaviors in diffusion models: (i) MtoG scaling behaviors with model capacity and training size (Appendix B.1), (ii) early learning and double descent in learning dynamics (Appendix B.2), and (iii) bias-variance trade-off of generalization error (Appendix B.3).

B.1. Scaling Behaviors of the MtoG Transition

First, we investigate the scaling behavior of the MtoG transition with respect to both model capacity $|\boldsymbol{\theta}|$ and training data size N, using the metrics $\mathcal{E}_{\texttt{gen}}$ and $\mathcal{E}_{\texttt{mem}}$. We evaluate ten U-Net architectures on the CIFAR-10 dataset, with model sizes ranging from 0.9M to 55.7M parameters (U-Net-1 to U-Net-10). For each model, we compute $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ across varying training dataset sizes, following the evaluation protocol outlined in Section 3. We report our results in Figure 1 with additional experimental details provided in Appendix D.4, where we observe the following:

Finding I.1: Scaling training data N induces MtoG transition under fixed model capacity $|\boldsymbol{\theta}|$. As shown in Figure 1 (left), for a fixed model capacity $|\boldsymbol{\theta}|$, our metrics reveal a clear transition from memorization to generalization as the number of training samples N increases. Notably,

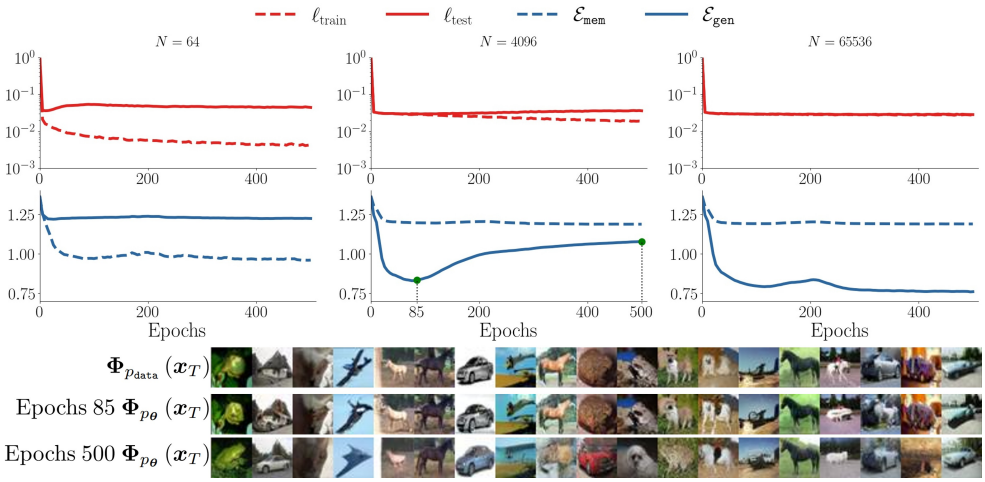

Figure 2: **Training dynamics of diffusion models in different regimes.** The top figure plots $\mathcal{E}_{\texttt{mem}}, \mathcal{E}_{\texttt{gen}}, \ell_{\texttt{train}}, \ell_{\texttt{test}}$ over training epochs for different different dataset sizes: $N = 2^6$ (left), 2^{12} (middle), 2^{16} (right). The bottom figure visualizes the generation when $N = 2^{12}$. The top row shows samples from the underlying distribution $\mathbf{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T)$, while the middle and bottom rows display outputs from the trained diffusion model $\mathbf{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$ at epoch 85 and 500, respectively.

larger models transition more slowly to generalization, as their greater capacity allows them to memorize more of the training data. Compared to prior studies of this transition [73, 75], our results more accurately capture the underlying behavior by directly measuring the distributional distance between the learned and ground-truth distributions. In contrast, earlier approaches [73, 75] assess generalization based on the deviation of generated samples from the training data, which does not reliably reflect true generalization.

Finding I.2: MtoG transition governed consistently by the ratio $N/\sqrt{|\boldsymbol{\theta}|}$. Moreover, in contrast to prior work that focuses solely on the effect of training sample size N, our results in Figure 1 (right) reveal a consistent scaling behavior when using our metric, governed by the ratio $N/\sqrt{|\boldsymbol{\theta}|}$ between data size and model capacity. Remarkably, both $\mathcal{E}_{\texttt{gen}}$ and $\mathcal{E}_{\texttt{mem}}$ metrics exhibit near-identical MtoG transition curves across models of varying sizes when plotted against this ratio. As such, analogous to the empirical scaling laws observed in large language models [35], this predictable trend provides practical guidance for the development of diffusion models, particularly when scaling up model size, data, or compute to achieve optimal performance gains.

B.2. Early Learning and Double Descent in Learning Dynamics

Building on the findings in Appendix B.1, we further examine the generalization behavior across different training regimes. Under the evaluation protocol in Section 3, we analyze the learning dynamics of a U-Net model with fixed model capacity (UNet-10 introduced in Appendix D.1) trained with the number of data samples $N = 2^6$, 2^{12}, and 2^{16}, corresponding to the memorization, transition, and generalization regimes in Appendix B.1, respectively. The model is trained using stochastic gradient descent (SGD) for 500 epochs, during which we track $\mathcal{E}_{\texttt{mem}}$, $\mathcal{E}_{\texttt{gen}}$, $\ell_{\texttt{train}}$, and $\ell_{\texttt{test}}$ at each epoch. The results in Figure 2 reveal several notable generalization behaviors that align with phenomena previously observed in the training of overparameterized deep models [52, 74]:

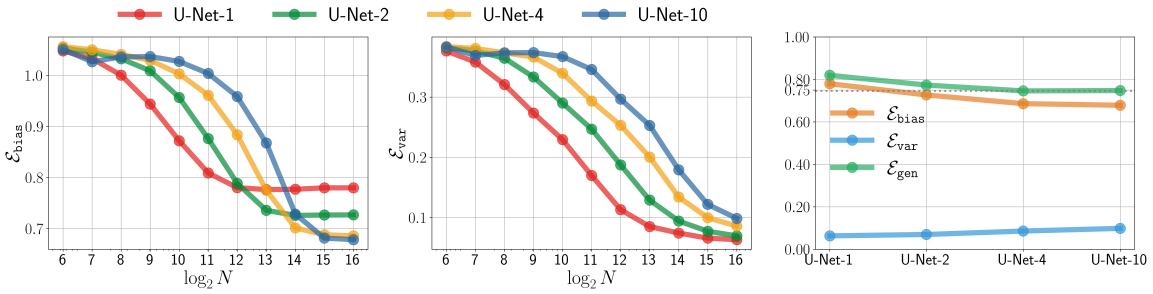

Figure 3: **Bias–Variance Trade-off.** Left figure shows $\mathcal{E}_{\texttt{bias}}$ and $\mathcal{E}_{\texttt{var}}$ as functions of the number of training samples N for various network architectures. Right figure plots the generalization error $\mathcal{E}_{\texttt{gen}}$, bias $\mathcal{E}_{\texttt{bias}}$, and variance $\mathcal{E}_{\texttt{var}}$ across different network architectures with a fixed training sample size of $N = 2^{16}$.

Finding II.1: Early learning behavior in memorization and transition regimes. As shown in Figure 2 (left & middle), in both the memorization ($N = 2^6$) and transition ($N = 2^{12}$) regimes, the generalization error initially decreases during training but reaches its minimum at an early epoch, after which it begins to increase again. This *early learning* (or early generalization) phenomenon becomes more salient as the training sample size increases from the memorization to the transition regime. As shown in the visualization at the bottom of Section 3, the model at Epoch 85 clearly exhibits generalization, whereas the model at Epoch 500 fails to generalize. This is also corroborated by the divergence of training loss $\ell_{\texttt{train}}$ and test loss $\ell_{\texttt{test}}$ at the top of the figure. It is worth mentioning that, although early learning behavior has been theoretically and visually demonstrated in previous works [44, 45], PFD is the first metric to provide empirical evidence of this phenomenon.
Finding II.2: Double descent of the generalization error in the generalization regime. In contrast, as shown in Figure 2 (right), training in the generalization regime ($N = 2^{16}$) reveals a clear instance of the *double descent* phenomenon [52] in the generalization error. Specifically, the error initially decreases, then increases during intermediate training epochs, and finally decreases again as training approaches convergence. Notably, this non-monotonic behavior is not captured by the standard training and test losses $\ell_{\texttt{train}}$ and $\ell_{\texttt{test}}$, both of which decrease monotonically throughout training. This implies that extended training can improve generalization performance in the generalization regime.
Remarks. For both cases, it should be noted that these generalization phenomena observed through our metrics are not unique to diffusion models. Similar surprising behaviors have been previously reported in training overparameterized deep learning models with extensive theoretical investigations [4, 5, 7, 52, 74]. For example, the early learning phenomenon has been widely observed when training models with limited or noisy data, such as in deep image priors [69] and learning with label noise [48, 49]. Similarly, the double descent phenomenon has been reported in the training dynamics of overparameterized models [52]. These observations challenge the traditional view of generalization and highlight the critical role of inductive bias and training time in the learning process. Similarly, our findings imply that such factors should also be carefully considered when training diffusion models.

B.3. Bias-variance Trade-off of the Generalization Error

In statistical learning theory, bias-variance trade-off is a classical yet fundamental concept in supervised learning which helps us understand and analyze the sources of prediction error in the model [6, 26, 38, 72]. Specifically, bias–variance decomposition expresses the expected generalization error as the sum of two components: (i) the *bias term*, which quantifies the discrepancy between the expected model prediction and the true function—high bias indicates systematic error or underfitting; and (ii) the *variance term*, which measures the prediction variability of the model across different training sets—high variance reflects sensitivity to data fluctuations or overfitting.

However, in unsupervised learning settings such as diffusion models, the notion of generalization error was not well-defined prior to our work, in contrast to the well-established definitions in supervised learning. As a result, bias–variance decomposition in this context remains largely unexplored. In this work, we address this gap through the generalization error measure $\mathcal{E}_{\texttt{gen}}$ (see Equation (9)), which admits a bias–variance decomposition analogous to that in the supervised setting, as we detail below.

Definition 3 (Bias-Variance Decomposition of $\mathcal{E}_{\texttt{gen}}$) *Based on the same setup as Definition 2, we can decompose $\mathcal{E}_{\texttt{gen}}$ in Equation (9) as*

$$\mathbb{E}_{\mathcal{D}}\left[\mathcal{E}^2_{\texttt{gen}}\left(p_{\boldsymbol{\theta}(\mathcal{D})}\right)\right] = \mathcal{E}^2_{\texttt{bias}} + \mathcal{E}_{\texttt{var}} \tag{12}$$

where $p_{\boldsymbol{\theta}(\mathcal{D})}$ denotes the distribution induced by a diffusion model $\boldsymbol{\theta}(\mathcal{D})$ trained on a given training dataset $\mathcal{D}$ sampled from $p_{\texttt{data}}$. Specifically, the bias and variance terms are defined as:

$$\mathcal{E}_{\texttt{bias}} := \mathbb{E}_{\boldsymbol{x}_T}([\|\boldsymbol{\Psi}\circ\boldsymbol{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T) - \overline{\boldsymbol{\Psi}\circ\boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)\|_2^2])^{1/2}, \mathcal{E}_{\texttt{var}} := \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\boldsymbol{x}_T}[\|\boldsymbol{\Psi}\circ\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}(\mathcal{D})}}(\boldsymbol{x}_T) - \overline{\boldsymbol{\Psi}\circ\boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)\|_2^2],$$

with $\overline{\boldsymbol{\Psi}\circ\boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\cdot) := \mathbb{E}_{\mathcal{D}}[\boldsymbol{\Psi}\circ\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}(\mathcal{D})}}(\cdot)]$.

Intuitively, our definitions of the bias term $\mathcal{E}_{\texttt{bias}}$ and the variance term $\mathcal{E}_{\texttt{var}}$ are both well-justified: (i) $\mathcal{E}_{\texttt{bias}}$ quantifies the systematic error between the learned distribution $p_{\boldsymbol{\theta}}$ and the ground-truth distribution $p_{\texttt{data}}$; and (ii) $\mathcal{E}_{\texttt{var}}$ captures the variability of model predictions across different training sets by measuring the distance between $p_{\boldsymbol{\theta}}$ and the mean $\overline{p_{\boldsymbol{\theta}}}$ which can be empirically estimated by averaging over multiple datasets $\mathcal{D}$ sampled from $p_{\texttt{data}}$. Experimental results, following the protocol in Section 3, are shown in Figure 3, with detailed settings in Appendix D.6.

In Figure 3 (a), when diffusion models are trained in the generalization regime, the resulting generalization decomposition aligns with classical bias–variance theory from supervised learning: as model complexity increases, the bias term $\mathcal{E}_{\texttt{bias}}$ decreases while the variance term $\mathcal{E}_{\texttt{var}}$ increases, resulting in a U-shaped generalization error curve. Additionally, Figure 3 (b) further illustrates the effect of the training sample size N and number of parameters $|\boldsymbol{\theta}|$: increasing N reduces both $\mathcal{E}_{\texttt{bias}}$ and $\mathcal{E}_{\texttt{var}}$, thereby lowering the generalization error $\mathcal{E}_{\texttt{gen}}$, as expected; In contrast, increasing $|\boldsymbol{\theta}|$ consistently increases $\mathcal{E}_{\texttt{var}}$, and its effect on $\mathcal{E}_{\texttt{bias}}$ depends on the size of N: it decreases $\mathcal{E}_{\texttt{bias}}$ when $N \geq 2^{15}$ but increases it when $N \leq 2^{11}$.

Appendix C. Proof in Section 2

Proof [Proof of Theorem 1] It is trivial to show $\texttt{PFD}(p,q) > 0$ for any $p \neq q$ and $\texttt{PFD}(p,q) = \texttt{PFD}(q,p)$, and thus we omit the proof.

- Proof of $p = q \Leftrightarrow \texttt{PFD}(p, q) = 0$:
  - ($\Rightarrow$) If $p = q$, $\nabla \log p_t(\boldsymbol{x}_t) = \nabla \log q_t(\boldsymbol{x}_t)$, thus:

$$\mathrm{d}\boldsymbol{x}_t = -t\left(\nabla \log p_t(\boldsymbol{x}_t) - \nabla \log q_t(\boldsymbol{x}_t)\right)\mathrm{d}t = 0 \tag{13}$$

    Thus, $\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_q(\boldsymbol{x}_T)$ is the solution of the ODE function Equation (13) with initial $\boldsymbol{x}_T = \mathbf{0}$. Thus $\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_1(\boldsymbol{x}_T) = \mathbf{0}$ for all $\boldsymbol{x}_T$. Thus $\texttt{PFD}(p, q) = 0$
  - ($\Leftarrow$) If $\texttt{PFD}(p, q) = 0$ and $\boldsymbol{\Phi}_p, \boldsymbol{\Phi}_q$ are continuous function w.r.t $\boldsymbol{x}_T$, then we have $\boldsymbol{\Phi}_p(\boldsymbol{x}_T) = \boldsymbol{\Phi}_q(\boldsymbol{x}_T)$ for all $\boldsymbol{x}_T$. If $\boldsymbol{x}_0 = \boldsymbol{\Phi}(\boldsymbol{x}_T)$, from the transformation of probability identities, we have:

$$p(\boldsymbol{x}_0) = \frac{\partial}{\partial[\boldsymbol{x}_0]_1} \cdots \frac{\partial}{\partial[\boldsymbol{x}_0]_n} \int_{\{\boldsymbol{\epsilon} \mid \boldsymbol{\Phi}(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0\}} p_{\mathcal{N}}(\boldsymbol{\epsilon}) \mathrm{d}^n \boldsymbol{\epsilon}, \tag{14}$$

    where $[\boldsymbol{x}_0]_i$ denotes the i-th element of $\boldsymbol{x}_0$, $\boldsymbol{f}(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0$ denotes the element wise less or equal. $p_{\mathcal{N}}(\cdot)$ is the probability density function (PDF) of Gaussian distribution $\mathcal{N}\left(\mathbf{0}, T^2 \boldsymbol{I}_n\right)$. Thus, for all $\boldsymbol{x}_0$ we have:

$$\begin{aligned}
p(\boldsymbol{x}_0) - q(\boldsymbol{x}_0) &= \frac{\partial}{\partial[\boldsymbol{x}_0]_1} \cdots \frac{\partial}{\partial[\boldsymbol{x}_0]_n} \int_{\{\boldsymbol{\epsilon} \mid \boldsymbol{\Phi}_p(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0\}} p_{\mathcal{N}}(\boldsymbol{\epsilon}) \mathrm{d}^n \boldsymbol{\epsilon} \\
&\quad - \frac{\partial}{\partial[\boldsymbol{x}_0]_1} \cdots \frac{\partial}{\partial[\boldsymbol{x}_0]_n} \int_{\{\boldsymbol{\epsilon} \mid \boldsymbol{\Phi}_q(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0\}} p_{\mathcal{N}}(\boldsymbol{\epsilon}) \mathrm{d}^n \boldsymbol{\epsilon}, \\
&= \frac{\partial}{\partial[\boldsymbol{x}_0]_1} \cdots \frac{\partial}{\partial[\boldsymbol{x}_0]_n} \int_{\{\boldsymbol{\epsilon} \mid \boldsymbol{\Phi}_p(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0\}} p_{\mathcal{N}}(\boldsymbol{\epsilon}) \mathrm{d}^n \boldsymbol{\epsilon} \\
&\quad - \frac{\partial}{\partial[\boldsymbol{x}_0]_1} \cdots \frac{\partial}{\partial[\boldsymbol{x}_0]_n} \int_{\{\boldsymbol{\epsilon} \mid \boldsymbol{\Phi}_p(\boldsymbol{\epsilon}) \leq \boldsymbol{x}_0\}} p_{\mathcal{N}}(\boldsymbol{\epsilon}) \mathrm{d}^n \boldsymbol{\epsilon}, \\
&= 0,
\end{aligned} \tag{15}$$

    so $p = q$.

- Proof of $\texttt{PFD}(p, q) \leq \texttt{PFD}(p, p') + \texttt{PFD}(p', q)$:

$$\begin{aligned}
&\texttt{PFD}(p, q) \\
&= \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2 \boldsymbol{I})}\left[\left\|\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_q(\boldsymbol{x}_T)\right\|_2^2\right]\right)^{1/2} \\
&\leq \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2 \boldsymbol{I})}\left[\left(\left\|\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_{p'}(\boldsymbol{x}_T)\right\|_2 + \left\|\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_{p'}(\boldsymbol{x}_T)\right\|_2\right)^2\right]\right)^{1/2} \\
&\leq \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2 \boldsymbol{I})}\left[\left\|\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_q(\boldsymbol{x}_T)\right\|_2^2\right]\right)^{1/2} \\
&\quad + \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2 \boldsymbol{I})}\left[\left\|\boldsymbol{\Phi}_p(\boldsymbol{x}_T) - \boldsymbol{\Phi}_q(\boldsymbol{x}_T)\right\|_2^2\right]\right)^{1/2} \\
&= \texttt{PFD}(p, p') + \texttt{PFD}(p', q)
\end{aligned} \tag{16}$$

■

Lemma 1 *Under Assumption 1, for all $\boldsymbol{x}_T \in \mathcal{N}(\mathbf{0}, T^2\boldsymbol{I}_n)$, as $T \to \infty$, we have:*

$$\left\|\boldsymbol{\Phi}_p\left(\boldsymbol{x}_T\right)-\boldsymbol{\Phi}_q\left(\boldsymbol{x}_T\right)\right\|_2 \leq \exp\left(\frac{LT_\xi^2}{2}\right)\xi+\frac{\epsilon}{L}\left(\exp\left(\frac{LT_\xi^2}{2}\right)-1\right), \tag{17}$$

where ξ is a numerical constant and a finite timestep T_ξ depending only on ξ.

Proof [Proof of Theorem 1] Let $\boldsymbol{\phi}_t, t \in [0, T]$ denotes the ODE trajectory:

$$\begin{aligned}
\boldsymbol{\phi}_t &= \boldsymbol{x}_t^p - \boldsymbol{x}_t^q, \\
\boldsymbol{x}_t^p &= \boldsymbol{x}_T - \int_T^t \tau \nabla_{\boldsymbol{x}} \log p_\tau(\boldsymbol{x}_\tau^p) \mathrm{d}\tau, \\
\boldsymbol{x}_t^q &= \boldsymbol{x}_T - \int_T^t \tau \nabla_{\boldsymbol{x}} \log q_\tau(\boldsymbol{x}_\tau^q) \mathrm{d}\tau,
\end{aligned} \tag{18}$$

From the definition, $\boldsymbol{\phi}_0 = \boldsymbol{\Phi}_p\left(\boldsymbol{x}_T\right) - \boldsymbol{\Phi}_q\left(\boldsymbol{x}_T\right)$. Because $\lim_{T\to\infty} \boldsymbol{\phi}_t = \boldsymbol{x}_T - \boldsymbol{x}_T = \mathbf{0}$, from the $\epsilon - \delta$ definition of the limit, given $\boldsymbol{x}_T$, and a constant ξ , there exists a finite T_ξ related to ξ such that:

$$\left\|\boldsymbol{\phi}_t\right\|_2 \leq \xi \quad \text{for all } t \geq T_\xi. \tag{19}$$

As $t \leq T_\xi$, we have:

$$\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\phi}_t}{\mathrm{d}t} = -t\left(\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t^p) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t^q)\right), \\
&\left\|\boldsymbol{\phi}_{T_0}\right\|_2 \leq \xi.
\end{aligned} \tag{20}$$

Apply Assumption 1 to Equation (20), we could obtain the following integral inequality w.r.t $\left\|\boldsymbol{\phi}_t\right\|_2$:

$$\begin{aligned}
\frac{\mathrm{d}\left\|\boldsymbol{\phi}_t\right\|_2}{\mathrm{d}t} &\leq \left\|\frac{\mathrm{d}\boldsymbol{\phi}_t}{\mathrm{d}t}\right\|_2 \\
&\leq t\left\|\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t^p) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t^q)\right\|_2 \\
&\leq t\left(\epsilon + L\left\|\boldsymbol{\phi}_t\right\|_2\right), \\
\left\|\boldsymbol{\phi}_{T_\xi}\right\|_2 &\leq \xi, \quad 0 \leq t \leq T_\xi,
\end{aligned} \tag{21}$$

where the first inequality comes from the fact that $\frac{\mathrm{d}\left\|\boldsymbol{\phi}_t\right\|_2}{\mathrm{d}t} \leq \left\|\frac{\mathrm{d}\boldsymbol{\phi}_t}{\mathrm{d}t}\right\|_2$. From Grönwall's inequality [19], we could solve $\left\|\boldsymbol{\Phi}_p\left(\boldsymbol{x}_T\right) - \boldsymbol{\Phi}_q\left(\boldsymbol{x}_T\right)\right\|_2 = \left\|\boldsymbol{\phi}_0\right\|_2 \leq \exp(\frac{LT_\xi^2}{2})\xi + \frac{\epsilon}{L}\left(\exp(\frac{LT_\xi^2}{2}) - 1\right)$. ■

Proof [Proof of Theorem 2]Let $\boldsymbol{X} \coloneqq \left\|\boldsymbol{\Phi}_p\left(\boldsymbol{x}_T\right) - \boldsymbol{\Phi}_q\left(\boldsymbol{x}_T\right)\right\|_2^2$. From Theorem 1,

$$0 \leq \boldsymbol{X} \leq \kappa^2\left(L, \epsilon\right),$$

with $\kappa\left(L, \epsilon\right) \coloneqq \exp\left(\frac{LT_\xi^2}{2}\right)\xi + \frac{\epsilon}{L}\left(\exp\left(\frac{LT_\xi^2}{2}\right) - 1\right)$. From Hoeffding's inequality [30], we have:

$$\mathbb{P}\left(\left|\mathbb{E}[\boldsymbol{X}] - \frac{1}{M}\sum_{i=1}^M \boldsymbol{X}_i\right| \geq \gamma\right) \leq 2\exp\left(-\frac{2M\gamma^2}{\kappa^4\left(L, \epsilon\right)}\right), \tag{22}$$

with M samples to achieve γ accuracy. Thus, we could guarantee $\mathbb{P}\left(\left|\mathbb{E}[\boldsymbol{X}]-\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{X}_i\right|\leq\gamma\right)$ with probability η, when:

$$M\geq\frac{\kappa^4\left(L,\epsilon\right)}{2\gamma^2}\log\frac{2}{\eta}. \tag{23}$$

Because

$$\left|\mathtt{PFD}(p,q)-\hat{\mathtt{PFD}}(p,q)\right|=\left|\sqrt{\mathbb{E}[\boldsymbol{X}]}-\sqrt{\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{X}_i}\right| \tag{24}$$

$$\leq\sqrt{\left|\mathbb{E}[\boldsymbol{X}]-\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{X}_i\right|}. \tag{25}$$

We could guarantee that $\mathbb{P}\left(\left|\mathtt{PFD}(p,q)-\hat{\mathtt{PFD}}(p,q)\right|\leq\gamma\right)$ with probability η, when:

$$M\geq\frac{\kappa^4\left(L,\epsilon\right)}{2\gamma^4}\log\frac{2}{\eta}. \tag{26}$$

■

Example 1 *The Wasserstein-2 distance* $\mathrm{W}_2(\cdot,\cdot)$ *is the lower bound of the probability flow distance, i.e.,*

$$\mathrm{W}_2(p,q)\leq\mathtt{PFD}(p,q), \tag{27}$$

Specifically, let p *and* q *be multivariate Gaussian distribution* $\mathcal{N}\left(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1\right)$, $\mathcal{N}\left(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2\right)$, *respectively, where* $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\in\mathbb{R}^n$ *and* $\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2\in\mathbb{R}^{n\times n}$. *The* PFD *is given by*

$$\mathtt{PFD}\left(p,q\right)=\left(\left\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\right\|_2+\left\|\boldsymbol{\Sigma}_1^{1/2}-\boldsymbol{\Sigma}_2^{1/2}\right\|_F\right)^{1/2}, \tag{28}$$

under this case, the equality in Equation (27) *holds when* $\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1$.

Proof [Proof of Example 1]

Proof of $W_2(p,q)\leq\mathtt{PFD}(p,q)$. From the definition of Wasserstein-2 distance:

$$W_2(p,q)=\inf_{\gamma\in\Gamma(p,q)}\left(\mathbb{E}_{(\boldsymbol{x}_p,\boldsymbol{x}_q)\sim\gamma}\left\|\boldsymbol{x}_p-\boldsymbol{x}_q\right\|_2^2\right)^{1/2}, \tag{29}$$

where $\Gamma\left(p,q\right)$ is the set of all couplings of p and q. As proofed by [66], the noise-to-image mapping $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Phi}_q$ pushes the Gaussian distribution $\mathcal{N}\left(\boldsymbol{0},T^2\boldsymbol{I}_n\right)$ to the p and q distribution respectively. Thus we could find the coupling $\gamma_{\mathtt{PFD}}\coloneqq\left(\boldsymbol{\Phi}_p,\boldsymbol{\Phi}_q\right)_{\#}\mathcal{N}\left(\boldsymbol{0},T^2\boldsymbol{I}_n\right)$, i.e., the pushforward of $\mathcal{N}\left(\boldsymbol{0},T^2\boldsymbol{I}_n\right)$ by $\left(\boldsymbol{\Phi}_p,\boldsymbol{\Phi}_q\right)$, such that

$$\mathtt{PFD}(p,q)=\left(\mathbb{E}_{(\boldsymbol{x}_p,\boldsymbol{x}_q)\sim\gamma_{\mathtt{PFD}}}\left\|\boldsymbol{x}_p-\boldsymbol{x}_q\right\|_2^2\right)^{1/2}\geq W_2(p,q) \tag{30}$$

When distribution $p\left(\boldsymbol{x}\right)$ is Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu},\boldsymbol{\Sigma}\right)$ with $\boldsymbol{\mu}\in\mathbb{R}^n$ and $\boldsymbol{\Sigma}\in\mathbb{R}^{n\times n}$, from Equation (1) , we have $p_t\left(\boldsymbol{x}\right)$ is $\mathcal{N}\left(\boldsymbol{\mu},\boldsymbol{\Sigma}+\sigma_t^2\boldsymbol{I}_n\right)$, thus the score function could be caluclated as,

$$\nabla_{\boldsymbol{x}}\log p_t\left(\boldsymbol{x}\right)=\left(\boldsymbol{\Sigma}+t^2\boldsymbol{I}_n\right)^{-1}\left(\boldsymbol{\mu}-\boldsymbol{x}\right). \tag{31}$$

By plugging in Equation (31) to Equation (3), we could obtain the ODE equation w.r.t $\boldsymbol{x}$:

$$\mathrm{d}\boldsymbol{x} = -t\left(\boldsymbol{\Sigma} + t^2\boldsymbol{I}_n\right)^{-1}(\boldsymbol{\mu} - \boldsymbol{x})\,\mathrm{d}t, . \tag{32}$$

The above ODE equation has a close form solution:

$$\boldsymbol{x}_t = \boldsymbol{\mu} + \boldsymbol{U}\,\mathrm{diag}\left(\left[\sqrt{\frac{\lambda_1 + t^2}{\lambda_1 + T^2}}, \ldots, \sqrt{\frac{\lambda_n + t^2}{\lambda_n + T^2}}\right]\right)\boldsymbol{U}^\top(\boldsymbol{x}_T - \boldsymbol{\mu}) \tag{33}$$

where $\boldsymbol{U}, \lambda_k, k \in [n]$ are singlar value decomposition of $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} = \boldsymbol{U}\,\mathrm{diag}\left([\lambda_1, \ldots, \lambda_n]\right)\boldsymbol{U}^\top$. $\mathrm{diag}\left(\cdot\right)$ convert a vector in $\mathbb{R}^n$ into diagonal matrix $\mathbb{R}^{n\times n}$, and $\boldsymbol{x}_T \sim \mathcal{N}\left(\boldsymbol{0}, T^2\boldsymbol{I}_n\right)$. Let $\boldsymbol{x}_T = T\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}_n\right)$. As $t = 0$ and $T \to \infty$, we have:

$$\boldsymbol{x}_t = \left(\boldsymbol{I}_n - \boldsymbol{U}\,\mathrm{diag}\left(\left[\sqrt{\frac{\lambda_1 + t^2}{\lambda_1 + T^2}}, \ldots, \sqrt{\frac{\lambda_n + t^2}{\lambda_n + T^2}}\right]\right)\boldsymbol{U}^\top\right)\boldsymbol{\mu}, \tag{34}$$

$$+ \boldsymbol{U}\,\mathrm{diag}\left(\left[T\sqrt{\frac{\lambda_1 + t^2}{\lambda_1 + T^2}}, \ldots, T\sqrt{\frac{\lambda_n + t^2}{\lambda_n + T^2}}\right]\right)\boldsymbol{U}^\top\boldsymbol{x}_T, \tag{35}$$

$$= \boldsymbol{\mu} + \boldsymbol{U}\,\mathrm{diag}\left(\left[\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}\right]\right)\boldsymbol{U}^\top\boldsymbol{x}_T, \tag{36}$$

$$= \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{x}_T = \boldsymbol{\Phi}\left(\boldsymbol{x}_T\right). \tag{37}$$

Thus, plugging in Definition 1, we have:

$$\mathtt{PFD}\left(p, q\right) = \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2\boldsymbol{I})}\left[\left\|\boldsymbol{\Phi}_1\left(\boldsymbol{x}_T\right) - \boldsymbol{\Phi}_2\left(\boldsymbol{x}_T\right)\right\|_2^2\right]\right)^{1/2} \tag{38}$$

$$= \left(\mathbb{E}_{\boldsymbol{x}_T \sim \mathcal{N}(0, T^2\boldsymbol{I})}\left[\left\|\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_1^{1/2}\boldsymbol{x}_T - \boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_2^{1/2}\boldsymbol{x}_T\right\|_2^2\right]\right)^{1/2} \tag{39}$$

$$= \left(\left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right\|_2^2 + \left\|\boldsymbol{\Sigma}_1^{1/2} - \boldsymbol{\Sigma}_2^{1/2}\right\|_F^2\right)^{1/2} \tag{40}$$

$$= \left(\left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right)\right)^{1/2} \tag{41}$$

From Wasserstein-2 distance for Gaussian distribution p, q has close form solution:

$$W_2\left(p, q\right) = \left(||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 - 2\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{1/2}\right)^{1/2}\right)\right)^{1/2}. \tag{42}$$

From Theorem 2, we have $W_2\left(p, q\right) \leq \mathtt{PFD}\left(p, q\right)$. And specifically, $W_2\left(p, q\right) = \mathtt{PFD}\left(p, q\right)$ when $\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1$. ■

Lemma 2 *Given two positive semi-definite matrix $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^{n\times n}$,*

$$0 \leq \mathrm{Tr}\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right) \leq \mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{1/2}\right)^{1/2}\right). \tag{43}$$

Table 1: **U-Net architectures details.**

Name	Dimensions for encoder blocks	Number of residual blocks	Number of parameters $\lvert\boldsymbol{\theta}\rvert$
U-Net-1	[32, 32, 32]	4	0.9M
U-Net-2	[64, 64, 64]	4	3.5M
U-Net-3	[96, 96, 96]	4	7.9M
U-Net-4	[128, 128, 128]	4	14.0M
U-Net-5	[80, 160, 160]	4	17.1M
U-Net-6	[160, 160, 160]	3	17.8M
U-Net-7	[160, 160, 160]	4	21.8M
U-Net-8	[192, 192, 192]	4	31.3M
U-Net-9	[224, 224, 224]	4	42.7M
U-Net-10	[256, 256, 256]	4	55.7M

Proof [Proof of Theorem 2] Because $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ are positive semi-definite matrix, $\mathrm{Tr}\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right) \geq 0$ and

$$\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{1/2}\right)^{1/2}\right) = \mathrm{Tr}\left(\sqrt{\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right)\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right)^{\top}}\right) = \left\|\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right\|_*, \tag{44}$$

where $||\cdot||_*$ is the nuclear norm (trace norm). From trace norm inequality ([8] Chapter IV, Section 2), for a random matrix $\boldsymbol{M}$, $\mathrm{Tr}\left(\boldsymbol{M}\right) \leq \|\boldsymbol{M}\|_*$. Thus, we have:

$$\mathrm{Tr}\left(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right) \leq \left\|\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_2^{1/2}\right\|_*. \tag{45}$$

■

Appendix D. Experiments

In this section, we provide experimental details and additional discussion of the main results presented in Section 3 and Appendix B.

D.1. Network architecture details

In this subsection, we provide details of the U-Net architectures, as summarized in Table 1. The U-Net follows an encoder-decoder design, where the encoder comprises multiple encoder blocks. The column "**Dimensions for encoder blocks**" indicates the feature dimensions of each encoder block, while "**Number of residual blocks**" specifies how many residual blocks are used within each encoder block. The decoder is symmetric to the encoder. For further architectural details, please refer to [37]. By varying the encoder block dimensions and the number of residual blocks, we scale the U-Net model from 0.9M to 55.7M parameters.

D.2. Evaluation protocol

In this subsection, we provide details of the evaluation protocol introduced in Section 3, as well as the comparison between the synthetic data from the teacher model and the real dataset.

Experiment settings for evaluation protocol. The teacher model $\boldsymbol{\theta}_t$ and the student model $\boldsymbol{\theta}$ share a similar U-Net architecture [60] with different numbers of parameters, as introduced in Appendix D.1. The teacher model, with UNet-10 architecture, is trained on the CIFAR-10 dataset [39] using the EDM noise scheduler [36], with a batch size of 128 for 1,000 epochs. The student model [2] is trained using the variance-preserving (VP) noise scheduler [29], under the same training hyperparameters. We use one A40 GPU with 48 GB video random access memory (VRAM) for all experiments. We generated three subsets of initial noise $\{\boldsymbol{x}^{(i)}_{\texttt{train},T}\}_{i=1}^N, \{\boldsymbol{x}^{(i)}_{\texttt{gen},T}\}_{i=1}^M, \{\boldsymbol{x}^{(i)}_{\texttt{test},T}\}_{i=1}^M \overset{\text{iid}}{\sim} \mathcal{N}(0, T^2\boldsymbol{I}_n)$. The training and test datasets are produced using the teacher model:

$$\mathcal{D} := \{\boldsymbol{x}^{(i)}_{\texttt{train}}\}_{i=1}^N = \{\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}_t}}(\boldsymbol{x}^{(i)}_{\texttt{train},T})\}_{i=1}^N, \;\; \mathcal{D}_{\texttt{test}} := \{\boldsymbol{x}^{(i)}_{\texttt{test}}\}_{i=1}^M = \{\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}_t}}(\boldsymbol{x}^{(i)}_{\texttt{test},T})\}_{i=1}^M.$$

To evaluate the student model, we generate an evaluation dataset from itself:

$$\mathcal{D}_{\texttt{gen}} := \{\boldsymbol{x}^{(i)}_{\texttt{gen}}\}_{i=1}^M = \{\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}^{(i)}_{\texttt{gen},T})\}_{i=1}^M.$$

All samples are generated using the second-order Heun solver [36] with 18 sampling steps. We vary the number of training samples N from 2^6 to 2^{16} in powers of two. M is set to 50,000 for the experiments in Appendix D.3, and 10,000 for the rest.

Experiment settings for validating the synthetic dataset with real real-world dataset. We evaluate FID and $\mathcal{E}_{\texttt{mem}}$ for diffusion models with UNet-4 architecture, trained separately on the synthetic dataset $\mathcal{D}$ and CIFAR-10 training dataset. We keep the number of training dataset N the same for these two settings, ranging from 2^6 to 2^{15}, with a power of 2. Then we evaluate the FID between $\mathcal{D}_{\texttt{gen}}$ and $\mathcal{D}_{\texttt{test}}$ (CIFAR-10 test dataset) for the synthetic (real-world) setting, with $M = 10000$. To evaluate $\mathcal{E}_{\texttt{mem}}$, we use the initial noise $\{\boldsymbol{x}^{(i)}_{\texttt{gen}}\}_{i=1}^M$.

D.3. Comparison with practical metrics for generalization evaluation

In this subsection, we expand upon the experiment presented in Section 3, which compares our proposed metric with practical metrics for evaluating generalization. We compare $\mathcal{E}_{\texttt{gen}}$ and $\mathcal{E}_{\texttt{mem}}$ with well-used generative model metrics, inluding FID, $\texttt{FD}_{\texttt{DINOv2}}$, KID, CMMD, Precision, Recall, NND, IS. We also including the training and testing loss $\ell_{\texttt{train}}, \ell_{\texttt{test}}$ (Equation (10)) as comparison. We evaluating their ability in capturing the MtoG transition, under the evaluation protocol proposed in Section 3.

We use UNet-10 for the student model in this experiment. We summarized datasets used by these metrics in Table 2. Results are shown in Figure 4, summarized into one sentence, only $\mathcal{E}_{\texttt{gen}}$ and $\mathcal{E}_{\texttt{mem}}$ could quantitatively capture this transition. We include detailed discussions below:

Results discussions. Figure 4 (bottom) is consistent with prior empirical observations [73, 75]: In the memorization regimes ($N = 2^6$), $p_{\boldsymbol{\theta}}$ tends to memorize the empirical distribution $p_{\texttt{emp}}$, resulting in similar generation between $\boldsymbol{\Phi}_{p_{\texttt{emp}}}(\boldsymbol{x}_T)$ and $\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$; in the transition regime ($N = 2^{12}$), the model lacks sufficient capacity to memorize and the sample complexity is inadequate for generalization, leading to poor-quality generations $\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$; in the generalization regimes ($N = 2^{16}$), $p_{\boldsymbol{\theta}}$ captures the underlying distribution $p_{\texttt{data}}$, and the generations $\boldsymbol{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T)$ and $\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$ are closely aligned.

As shown in Figure 4 (top), when N increases, $\mathcal{E}_{\texttt{mem}}$ consistently increases and $\mathcal{E}_{\texttt{gen}}$ consistently decreases. This aligns with our intuition: as sample complexity grows, models tend to generalize and memorize less. In contrast, all other metrics fail to capture this transition effectively. The reasons can be summarized as follows:

2. The architecture of the student model varies across experiments and will be described in detail for each specific case.

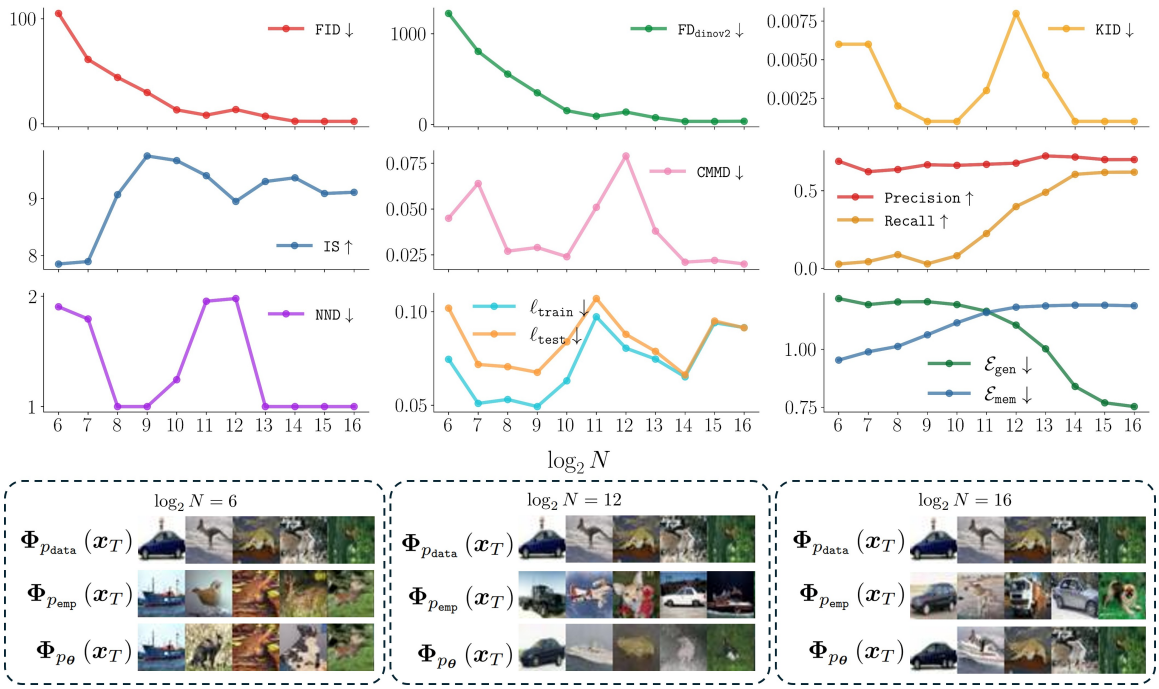

Figure 4: **Comparison of practical metrics on the MtoG transition.** The top figure plots multiple evaluation metrics as functions of $\log_2 N$. The bottom figure visualizes the generation under three numbers of training samples ($2^6, 2^{12}, 2^{16}$). For each setting, the figure shows generations from the underlying distribution (top row), empirical data distribution (middle row), and the learned distribution from the diffusion model (bottom row). Each column corresponds to the same initial noise.

Metric	**Dataset(s)**
FID, $\text{FD}_{\text{DINOv2}}$, KID, CMMD, Precision, Recall, NND	$\mathcal{D}_{\text{gen}}$ vs. $\mathcal{D}_{\text{test}}$
$\text{FID}_{\text{train}}$, $\text{FD}_{\text{DINOv2,train}}$	$\mathcal{D}$ vs. $\mathcal{D}_{\text{test}}$
IS	$\mathcal{D}_{\text{gen}}$
ℓ_{train}	$\mathcal{D}$
ℓ_{test}	$\mathcal{D}_{\text{test}}$
$\mathcal{E}_{\text{mem}}$, $\mathcal{E}_{\text{gen}}$	$\{\boldsymbol{x}^{(i)}_{\text{gen},T}\}_{i=1}^{M}$

Table 2: Datasets used to evaluate each metric.

- **FID, $\text{FD}_{\text{DINOv2}}$, KID, IS, and CMMD are sensitive to generation quality.** Image quality metrics, including FID, $\text{FD}_{\text{DINOv2}}$, KID, IS, and CMMD, show degradation in performance at $N = 2^{12}$. This drop is primarily due to degraded visual quality in the generated samples, as visualize in Figure 4 (bottom-middle). However, at this sample complexity, the generated data still captures low-level features such as colors and structures from the underlying distribution. This is evident

from the visual similarity between $\mathbf{\Phi}_{p_{\texttt{data}}}(\boldsymbol{x}_T)$ and $\mathbf{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$, suggesting the model have some generalizability. In comparison, only $\mathcal{E}_{\texttt{gen}}$ decreases consistently around $N = 2^{12}$, indicating it captures generalizability better than others despite visual degradation.

- **$\texttt{FID}$, $\texttt{FD}_{\texttt{DINOv2}}$ and $\texttt{Recall}$ are sensitive to diversity.** The monotonic trends for $\texttt{FID}$, $\texttt{FD}_{\texttt{DINOv2}}$ and $\texttt{Recall}$ are due to their sensitivity to the diversity of $\mathcal{D}_{\texttt{gen}}$, rather than their ability to measure generalizability. At small N, the model memorizes the training samples, resulting in $\mathcal{D}_{\texttt{gen}}$ closely resembling $\mathcal{D}$ and exhibiting significantly lower diversity than $\mathcal{D}_{\texttt{test}}$, since $N \ll M$. Under these conditions, $\texttt{FID}$, $\texttt{FD}_{\texttt{DINOv2}}$ are large because they are biased towards the diversity of the evaluation samples (as proved in [18]). Meanwhile, $\texttt{Recall}$ is low because the the support of $\mathcal{D}_{\texttt{test}}$ is limited, reducing the probability that samples drawn from $\mathcal{D}_{\texttt{gen}}$ lie within the support of $\mathcal{D}_{\texttt{test}}$. In contrast, $\mathcal{E}_{\texttt{gen}}$ measures generalizability by directly quantifying the distance between the generation from the learned distribution and the underlying distribution and is less affected by the diversity of the generated samples.

- **$\texttt{NND}$ and ℓ fail to capture the generalizability.** The $\texttt{NND}$, originally designed for assessing the generalization of GANs, is sensitive to image quality and increases during the transition regime. Additionally, it produces identical values across a wide range of sample sizes (e.g., $N = 2^8, 2^9, 2^{13}, 2^{14}, 2^{15}, 2^{16}$), making it unreliable for evaluating generalization in diffusion models. Similarly, neither the training loss $\ell_{\texttt{train}}$ nor the test loss $\ell_{\texttt{test}}$ exhibits a consistent decreasing trend as N increases, indicating that these losses do not directly reflect either memorization or generalization. While the loss gap $\ell_{\texttt{test}} - \ell_{\texttt{train}}$ does tend to decrease with larger N, it cannot serve as a robust generalization metric either. This is because even a randomly initialized model $\boldsymbol{\theta}$ can exhibit a small loss gap.

In conclusion, $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ are the only metrics that could capture the MtoG transition for diffusion models. They evaluate the generalization (memorization) by directly measuring the distance between the learned distribution by the diffusion model and the underlying (empirical) distribution. Unlike other metrics, they are less affected by the quality or diversity of the evaluating samples.

D.4. Scaling Behaviors of the MtoG Transition

In this subsection, we provide detailed experimental settings for Appendix B.1, along with additional experiments to further investigate the MtoG transition across more architectures (e.g., Transformer-based models [3]).

Experiment settings The detailed architectures of the student models, from U-Net-1 to U-Net-10, are provided in Appendix D.1, with model sizes ranging from 0.9M to 55.7M parameters. We scale up the architectures by increasing the dimensionality of the encoder blocks and the number of residual blocks.

MtoG Transition between U-Net and Transformer architecture To further investigate the impact of network architecture, we compare the U-Net architecture with the transformer-based UViT [3]. Specifically, we use the U-Net-9 from Table 1, containing 42.7M parameters, and design the UViT model with comparable parameters of 44.2M. Both models are trained for 1000 epochs. Using the same experimental setup described in Appendix B.1, we plot the MtoG transition curves for both U-Net and UViT, as shown in Figure 5.

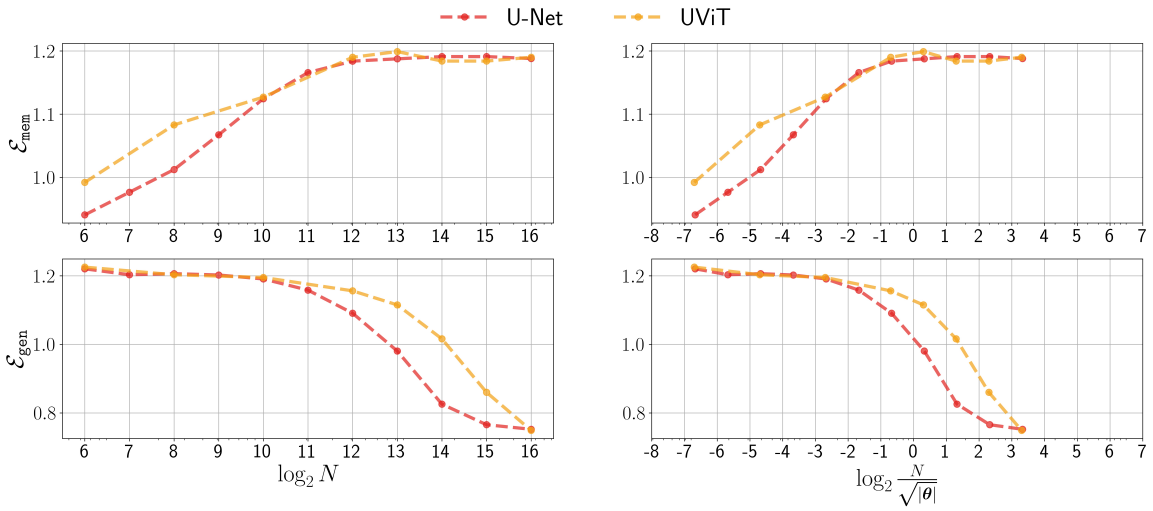

Figure 5: **Comparison of scaling behavior between UNet and Transformer architectures in the MtoG transition.** $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ plotted against $\log_2(N)$ for U-Net architecture (U-Net-9) and UViT architecture. Right: the same metrics plotted against $\log_2(N/\sqrt{|\boldsymbol{\theta}|})$, where $|\boldsymbol{\theta}|$ is the number of model parameters.

As illustrated in Figure 5, with a similar number of parameters and the same training data sizes, UViT exhibits a higher $\mathcal{E}_{\texttt{mem}}$ in the memorization regime ($2^6 \leq N \leq 2^{10}$) and a higher $\mathcal{E}_{\texttt{gen}}$ in the generalization regime ($2^{11} \leq N \leq 2^{15}$), suggesting a lower model capacity compared to U-Net under these conditions. However, when provided with sufficient training data ($N = 2^{16}$), UViT achieves a lower $\mathcal{E}_{\texttt{gen}}$, demonstrating better generalization performance. This observation is consistent with prior findings on transformer architectures in classification tasks: transformer-based models, lacking the inductive biases inherent to CNNs, tend to generalize poorly when trained on limited data [20].

D.5. Early Learning and Double Descent in Learning Dynamics

In this subsection, we build on the discussion from Appendix B.2. In Figure 2, we evaluate $\ell_{\texttt{train}}$ and $\ell_{\texttt{test}}$ across the three training regimes. Notably, the gap $\ell_{\texttt{test}} - \ell_{\texttt{train}}$ emerges as a practical heuristic for identifying the training regime: In the memorization regime, the gap increases steadily with training; In the transition regime, the gap remains near zero during early training (when generalization improves) and increases for further training (when generalization degrades); in the generalization regime, the gap remains close to zero throughout training. While $\ell_{\texttt{test}} - \ell_{\texttt{train}}$ is not a strict measure of generalization, it proves to be a useful empirical indicator of training regimes for diffusion models. Practically, by setting aside a test dataset to estimate this gap, we can more effectively identify the training regime for diffusion models.

D.6. Bias-variance Decomposition of Generalization Error

To approximate $\overline{\boldsymbol{\Psi} \circ \boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\cdot)$, we independently sample two training datasets, $\mathcal{D}_1$ and $\mathcal{D}_2$, for each specified number of training samples N. We then train two student models, $\boldsymbol{\theta}(\mathcal{D}_1)$ and $\boldsymbol{\theta}(\mathcal{D}_2)$, using these datasets. The quantity $\overline{\boldsymbol{\Psi} \circ \boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\cdot)$ is approximated as follows:

$$\overline{\boldsymbol{\Psi} \circ \boldsymbol{\Phi}}_{p_{\boldsymbol{\theta}}}(\cdot) \approx \frac{1}{2}(\boldsymbol{\Psi} \circ \boldsymbol{\Phi}_{p_{\boldsymbol{\theta}(\mathcal{D}_1)}}(\cdot) + \boldsymbol{\Psi} \circ \boldsymbol{\Phi}_{p_{\boldsymbol{\theta}(\mathcal{D}_2)}}(\cdot)). \tag{46}$$

Appendix E. Further Discussions of $\mathcal{E}_{\texttt{mem}}$

In this section, we present the mathematical formulation for estimating $\mathcal{E}_{\texttt{mem}}$ and compare it with existing memorization metric.

Empirically estimate $\mathcal{E}_{\texttt{mem}}$. As described in Definition 1 and Definition 2, estimating $\mathcal{E}_{\texttt{mem}}$ requires access to the mapping $\boldsymbol{\Phi}_{p_{\texttt{emp}}}(\cdot)$. According to Equation (3), this mapping is determined by the score function of the empirical distribution, denoted as $\nabla \log \hat{p}_t(\boldsymbol{x}_t)$. Based on prior works [25, 36, 74], the score function of the empirical distribution has a closed-form expression:

$$\nabla \log \hat{p}_t(\boldsymbol{x}_t) = \frac{1}{T^2}\left(\frac{\mathbb{E}_{\boldsymbol{x}\sim p_{\texttt{emp}}}[\mathcal{N}(\boldsymbol{x}_t;\boldsymbol{x},T^2\boldsymbol{I}_n)\cdot\boldsymbol{x}]}{\mathbb{E}_{\boldsymbol{x}\sim p_{\texttt{emp}}}[\mathcal{N}(\boldsymbol{x}_t;\boldsymbol{x},T^2\boldsymbol{I}_n)]} - \boldsymbol{x}_t\right), \tag{47}$$

where $p_{\texttt{emp}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{y}^{(i)})$ corresponds to the empirical distribution over the training dataset ${\boldsymbol{y}^{(i)}}_{i=1}^{N}$. This formulation allows us to numerically compute $\nabla \log \hat{p}_t(\boldsymbol{x}_t)$ for any given t. Subsequently, we can use a numerical solver to estimate the integral in Equation (3), thereby enabling the estimation of $\mathcal{E}_{\texttt{mem}}$.

Comparison between existing memorization metric and $\mathcal{E}_{\texttt{mem}}$. Previous works [73, 75] define memorization metirc as:

$$\texttt{M Distance}\,(p_{\boldsymbol{\theta}}) := \mathbb{E}_{\boldsymbol{x}_T}\left[\min_{\boldsymbol{x}\sim p_{\texttt{emp}}} \|\boldsymbol{\Psi}(\boldsymbol{x}) - \boldsymbol{\Psi}\circ\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)\|_2\right], \tag{48}$$

A generated sample $\boldsymbol{\Phi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_T)$ is a memorized sample if it is close enough to one of the sample $\boldsymbol{x}$ from $p_{\texttt{emp}}$. It is easy to show that $\mathcal{E}_{\texttt{mem}}$ is a more strict metric than `M Distance`, i.e. "$\mathcal{E}_{\texttt{mem}}\left(p^{\boldsymbol{\theta}}\right) = 0$" is a sufficient but not necessary condition for "`M Distance` $\left(p^{\boldsymbol{\theta}}\right) = 0$".

Appendix F. Ablation Study

In this section, we present ablation studies on the evaluation protocol, examining the effects of different noise schedulers and sampling methods (Appendix F.1), image descriptors (Appendix F.2), sample sizes for evaluation (Appendix F.3), and teacher models (Appendix F.4).

F.1. Sampling Methods

In this subsection, we present ablation studies on various noise schedulers and sampling strategies. Specifically, we evaluate the performance of the following methods: Variance Preserving (VP) [66], Variance Exploding (VE) [66], iDDPM [53] + DDIM [65], and EDM [36]. The specific form of $f(t), g(t)$ used in each approach are detailed in Table 1 of [36]. Additionally, each method also differs in its choice of ODE solver and timestep discretization strategy. For sampling, we use 256 steps for VP, 1000 for VE, 100 for iDDPM + DDIM, and 18 for EDM. All experiments are conducted under the evaluation protocol described in Section 3, where we estimate the $\mathcal{E}_{\texttt{gen}}$ under different

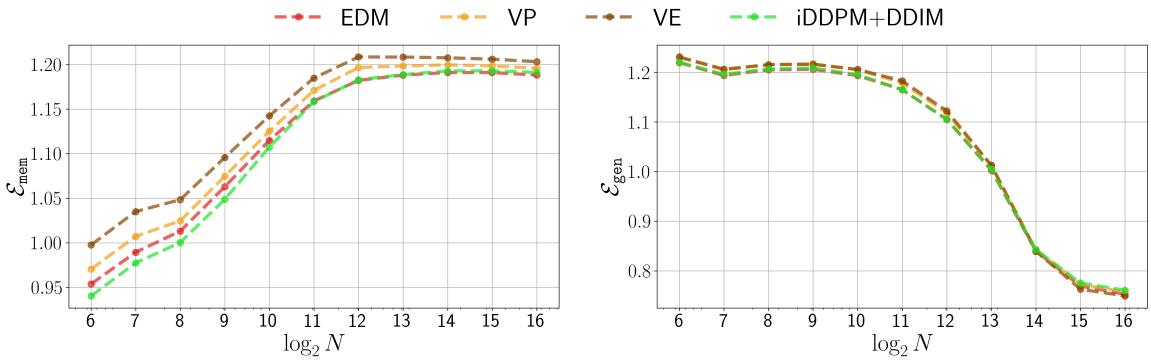

Figure 6: **Comparison of different sampling methods.** $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ plotted against $\log_2(N)$ for different sampling methods, including: EDM, VP, VE, iDDPM+DDIM.

training samples N. The student models use the UNet-10 architecture. During the ablation study, both the teacher and student models use the same sampling method[3] as specified above.

As shown in Figure 6, different samplers yield highly consistent results, demonstrating that PFD can be extended to various noise schedules, i.e., different choices of $f(t)$ and $g(t)$.

F.2. Image Descriptors

In this subsection, we present ablation studies on the image descriptor $\mathbf{\Psi}$ used in Equation (4). The descriptors evaluated include DINOv2 [56], InceptionV3 [68], CLIP [58], SSCD [57], and the identity function. All experiments follow the evaluation protocol described in Section 3, where we estimate both $\mathcal{E}_{\texttt{mem}}$ and $\mathcal{E}_{\texttt{gen}}$ across varying training sample sizes N and different student model architectures: U-Net-1, U-Net-2, U-Net-4, and U-Net-10.

As shown in Figure 7, different feature embeddings reveal a consistent trend in the memorization-to-generalization (MtoG) transition across various U-Net architectures. With limited training samples, smaller models exhibit lower generalization scores. Conversely, with sufficient training data, larger models tend to have lower generalization scores. When comparing with $\mathcal{E}_{\texttt{gen}}$ measured in pixel space (i.e., using the identity function as the descriptor), we observe that $\mathcal{E}_{\texttt{gen}}$ values are nearly identical across diffusion architectures when $N \geq 2^{15}$. In this regime, all models have learned low-level image features such as color and structure; however, only the larger models capture high-level perceptual details. Because pixel-space measurements fail to reflect these high-level features, they yield similar $\mathcal{E}_{\texttt{gen}}$ values regardless of model size. Therefore, it is better to evaluate $\mathcal{E}_{\texttt{gen}}$ in a feature space, which better captures perceptual differences between models.

F.3. Evaluation Sample Number

In this subsection, we present ablation studies on the number of samples M used by $\hat{\texttt{PFD}}$ to approximate PFD, as defined in Equation (5). All experiments follow the evaluation protocol described in Section 3, where we estimate $\mathcal{E}_{\texttt{gen}}$ across varying training sample sizes N and different student model architectures: U-Net-1, U-Net-2, U-Net-4, and U-Net-10. We vary $M \in$

3. Note that the noise scheduler used for sampling could differ from that used during training.

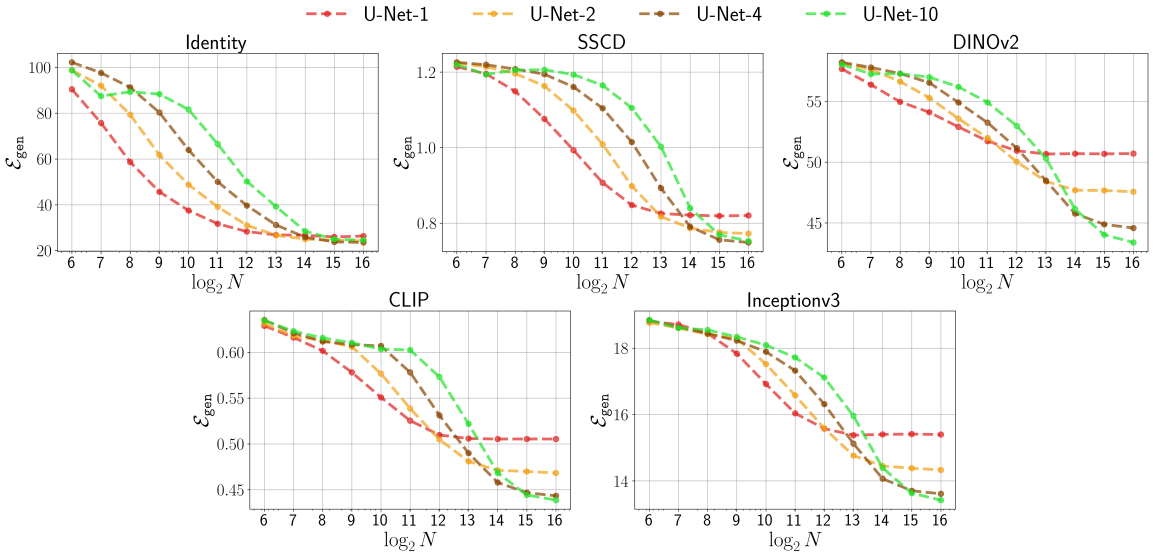

Figure 7: **Comparison between different image descriptors.** $\mathcal{E}_{\text{gen}}$ plotted against $\log_2(N)$ for a range of U-Net architectures (U-Net-1, U-Net-2, U-Net-4, U-Net-10) using different image descriptors, including identity function, SSCD, DINOv2, CLIP, Inceptionv3.

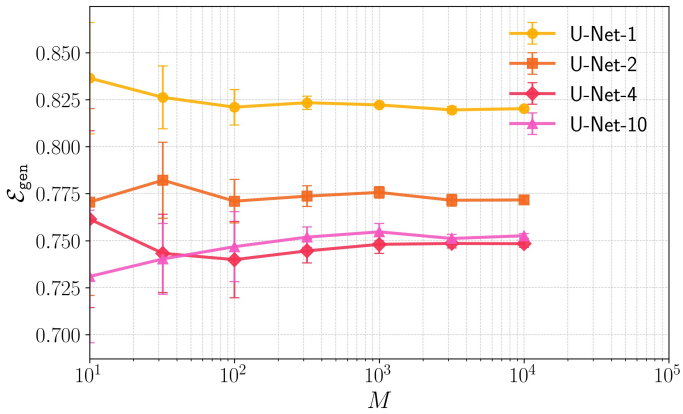

Figure 8: **Comparison across evaluation sample sizes.** The mean and variance of $\mathcal{E}_{\text{gen}}$ are plotted against the number of evaluation samples M for various U-Net architectures (U-Net-1, U-Net-2, U-Net-4, U-Net-10), with a fixed number of training samples $N = 2^{16}$.

$\{10, 32, 100, 316, 1000, 3163, 10000\}$, and for each setting, generate 5 independent sets of $\{\boldsymbol{x}^{(i)}_{\text{gen},T}\}_{i=1}^{M}$ initial noise estimate $\mathcal{E}_{\text{gen}}$, computing both the mean and variance. In this experiment, we use DINOv2 as the image descriptor.

As shown in Figure 8, the variance of $\mathcal{E}_{\text{gen}}$ approaches zero as M increases to 10,000, indicating that when $M \geq 10000$, the empirical estimate of $\mathcal{E}_{\text{gen}}$ converges to its value over the underlying distribution. This result holds consistently across different model architectures.

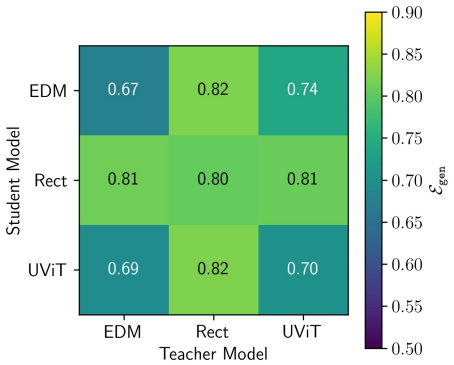

Figure 9: **Comparison of different teacher models.**The figure shows the $\mathcal{E}_{\texttt{gen}}$ values for various student models (EDM, Rect, UViT) trained using different teacher models (EDM, Rect, UViT), with a fixed training data size of $N = 2^{16}$.

F.4. Teacher Model Architecture

We end this section by examining how different teacher models affect the evaluation protocol. Specifically, we consider three types of diffusion models: EDM, Rectified Flow (Rect) [50], and UViT. Using the CIFAR-10 dataset, we train three teacher models, one for each of these diffusion types. For each teacher model, we then evaluate all three diffusion models as student models. We report their corresponding $\mathcal{E}_{\texttt{gen}}$ values. Both teacher and student models use the same sampling method, the second-order Heun solver with 18 steps.

As shown in Figure 9, the $\mathcal{E}_{\texttt{gen}}$ is approximately 0.7 when both the student and teacher models are selected from EDM or UViT. However, $\mathcal{E}_{\texttt{gen}}$ increases to around 0.8 when either the student or teacher model is Rect. According to its original paper, Rect has the poorest generation quality among the three, as measured by `FID`. This suggests that the teacher model should possess strong generative performance to serve as an underlying distribution that is close to the real-world data distribution. Therefore, in this paper, we adopt EDM as the teacher model, as it achieves the lowest `FID` among the three models.