Memorization effect of diffusion models

Yu-Han WU

Abstract

Diffusion models have achieved astonishing success in generative modeling in various tasks such as computer vision (Amit et al., 2021; Baranchuk et al., 2021), temporal data modeling (Alcaraz and Strodthoff, 2022; Chen et al., 2020), multi-modal modeling (Ramesh et al., 2022; Rombach et al., 2022), and natural language processing (NLP) (Austin et al., 2021; Savinov et al., 2021). The diffusion model for generative modeling was first proposed and studied by Sohl-Dickstein et al. (2015). Song and Ermon (2019) proposed score matching with Langevin dynamics and Ho et al. (2020) developed the Denoising Diffusion Probabilistic Models, which have become the major formulations of current research on diffusion models.

In this essay I start by introducing the mathematical theory of diffusion models. After the introduction of the main concepts behind diffusion models, I explain the memorization effect. Then, I show the empirical and theoretical result of the memorization effect. Finally, I point out some possible direction for future research.

1 Introduction to Diffusion Models

We make a slight abuse of notation by identifying a probability density and its distribution function for simplicity. Set π_{data} to be an unknown probability density on $\mathbb{R}^d$. The purpose of a generative model is the following: given some training data $X_1, \dots, X_n$ sampled from π_{data}, the goal is to generate new samples Y, mimicking drawings from π_{data}. The task of generative modeling has already been tackled by various approaches such as energy-based models (Teh et al., 2003; Du and Mordatch, 2019) or normalizing flows (Boffi and Vanden-Eijnden, 2022; Lipman et al., 2022). Diffusion models have emerged lately as the state-of-the-art approaches, achieving stunning performances in applications where traditional methods based on density-based estimation are proscribed given the dimensionality and the complexity of the data in play. The idea of diffusion models is the following: first in the "forward phase" we progressively add noise to the data and train a neural network to estimate the so-called score function simultaneously at each time step. This process transforms π_{data} into an easy-to-sample distribution π_∞, such as a Gaussian distribution. In addition, this phase may be seen as a discretization of an underlying continuous process, modeled by an stochastic differential equation (SDE). Then, in order to mimic samples from π_{data}, in the "backward phase" we generate new samples from π_∞ and reverse the previous transformation. In other words, we aim to find a series of distributions $(p_t)_{t\in[0,T]}$ with $p_0 = \pi_{\text{data}}$ and $p_T = \pi_\infty$ so that generating samples $X_t \sim p_t$ is doable.

1.1 Mathematical Description

Let $f : (t, x) \in [0, T] \times \mathbb{R}^d \mapsto f_t(x) \in \mathbb{R}^d$, $a : t \in [0, T] \mapsto a_t \in \mathbb{R}^d$ be smooth functions where $T \in \mathbb{R}_+$ is the terminal time. The forward SDE is defined as follows:

$$d\overrightarrow{X}_t = f_t(\overrightarrow{X}_t)dt + a_t d\overrightarrow{W}_t, \quad \overrightarrow{X}_0 \sim p_0, \tag{1}$$

It is shown in Anderson (1982) that the following backward SDE reverses the process in (1) in time:

$$d\overleftarrow{X}_t = -\big(f_{T-t}(\overleftarrow{X}_t) + a_{T-t}^2 \nabla \log p_{T-t}(\overleftarrow{X}_t)\big)dt + a_{T-t} d\overleftarrow{W}_t, \quad \overleftarrow{X}_0 \sim p_T. \tag{2}$$

In other words, the solution $\overleftarrow{X}$ in (2) satisfies $\overleftarrow{X}_{T-t} \sim \overrightarrow{X}_t$ for $t \in [0, T]$. It becomes clearer with Figure 1, where the top row represents the forward process, and the bottom row shows the backward process.

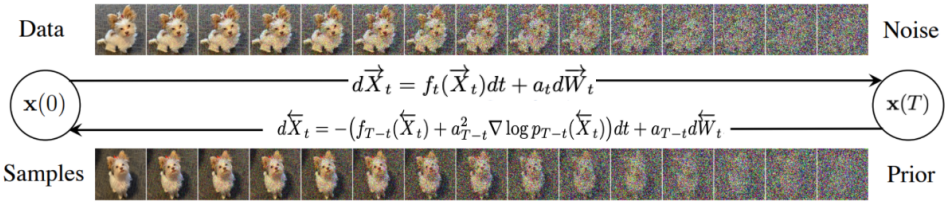

Figure 1: How sampling of a diffusion model works.

Additionally, it is also known that the following deterministic process with random initial condition

$$x'(t) = f_t(x(t)) - \frac{a_t^2}{2}\nabla \log p_t(x(t)), \quad x(0) = \overrightarrow{X}_0, \tag{3}$$

has the same marginal as in (1), i.e., $x(t) \sim p_t$. On the other hand, this process may also be reversed by solving the following equation:

$$y'(t) = -f_{T-t}(y(t)) + \frac{a_{T-t}^2}{2}\nabla \log p_{T-t}(y(t)), \quad y(0) \sim p_T. \tag{4}$$

In practice, the score function $\nabla \log p_t$ appearing in the drift term of the backward process is unknown, since it depends on unknown initial distribution π_{data}. Therefore, in order to solve (2) and (4), we have to estimate $\nabla \log p_t$, called the score function. This is often done by a procedure called score matching to have a parametrized neural network s_θ learning the score function $\nabla \log p_t$, we explain this concept in detail in the next section.

For simple functions f the process (1) has an explicit representation. Here we focus on the case where $f_t(x) = -b_t x$ for some function b, that is

$$dX_t = -b_t X_t + a_t d\overrightarrow{W}_t. \tag{5}$$

Define $\mu_t = \int_0^t b_s ds$. Then the solution of (5) is given by the following stochastic process

$$X_t = e^{-\mu_t} X_0 + \int_0^t e^{\mu_s - \mu_t} a_s dW_s.$$

In particular, the second term reduces to a Weiner integral, it is a centered Gaussian with variance $\int_0^t e^{2(\mu_s-\mu_t)} a_s^2 ds$, hence

$$X_t \overset{\text{law}}{=} e^{-\mu_t} X_0 + \mathcal{N}\left(0, \int_0^t e^{2(\mu_s-\mu_t)} a_s^2 ds\right).$$

A consequence of the preceding result is that when the variance $\sigma_t^2 = \int_0^t e^{2(\mu_s-\mu_t)} a_s^2 ds$ is significantly larger than $e^{-\mu_t}$, then the distribution p_t of $\overrightarrow{X}_t$ is well approximated by $\mathcal{N}(0, \sigma_t^2)$.

Therefore, in order to mimic new samples from the original data with the score function learnt with s_θ, we first generate new samples $Y_0 \sim \mathcal{N}(0, I)$, then with the learned score function s_θ we have the choice to simulate either the SDE

$$d\overleftarrow{X}_t = -\big(f_{T-t}(\overleftarrow{X}_t) + a_{T-t}^2 s_\theta(T-t, \overleftarrow{X}_t)\big)dt + a_{T-t} d\overleftarrow{W}_t, \quad \overleftarrow{X}_0 \sim \mathcal{N}(0, I_d). \tag{6}$$

to get samples Y_T or the ordinary differential equation (ODE)

$$y'(t) = -f_{T-t}(y(t)) + \frac{a_{T-t}^2}{2} s_\theta(T-t, y(t)), \quad y(0) \sim \mathcal{N}(0, I_d). \tag{7}$$

to get samples $y(T)$.

1.2 Score Matching

The $L2$-distance between the score of two probability densities is often called the Fisher divergence:

$$Fisher(\rho_1 \mid \rho_2) = \int \rho_1(x)\|\nabla \log \rho_1(x) - \nabla \log \rho_2(x)\|^2 dx.$$

As mentioned in the previous section, in order to generate samples with either (2) or (4), our goal is to choose a parametrized family of functions s_θ and to optimize θ so that the divergence

$$\int p_t(x)\|\nabla \log p_t(x) - s_\theta(x)\|^2 dx$$

is as small as possible. However, this optimization problem is intractable, due to the presence of the explicit form of p_t inside the integral. This is where Score Matching techniques come into play.

Vanilla Score Matching. Let p be a smooth probability density function supported over $\mathbb{R}^d$ and let X be a random variable with density p. The following result inspired the objective loss function of score matching.

Theorem 1 (Hyvärinen (2005); Vincent (2011)). *Let $s : \mathbb{R}^d \to \mathbb{R}^d$ be any smooth function with sufficiently fast decay to zero at ∞, and $X \sim p$. Then,*

$$\mathbb{E}\Big[\|\nabla \log p(X) - s(X)\|^2\Big] = c + \mathbb{E}\Big[\|s(X)\|^2 + 2\nabla \cdot s(X)\Big],$$

where c is a constant independent of s.

Therefore, in order to estimate the density function of the diffusion model, we use the following loss function

$$\ell(\theta) = \frac{1}{n}\mathbb{E}_{\tau\sim\rho}\Big[\sum_{i=1}^{n}\big(\|s_\theta(\tau, X_i(\tau))\|^2 + 2\nabla \cdot (s_\theta(\tau, X_i(\tau)))\big)\Big], \tag{8}$$

where ρ is some discrete distribution on $[0, T]$, and $X_i(\tau)$ are transformations of X_i by the forward process at time step τ. In the preceding formulation we cannot exactly compute the expectation with respect to p_t, but we can approximate it with our samples $X_i(t)$. Additionnaly, we need to approximate the integral, for instance we can discretize the time steps with $0 = t_0 < t_1 < \cdots < t_N = T$. Our objective function becomes

$$\ell(\theta) = \frac{1}{n}\sum_{t\in\{t_0,\ldots,t_N\}} w(t)\sum_{i=1}^{n}\Big(\|s_\theta(t, X_i(t))\|^2 + 2\nabla \cdot s_\theta(t, X_i(t))\Big)$$

Although being computable, this loss function is not ideal since in order to calcaulte the loss function, each time we do not only have to evaluate s_θ and its gradient but also the gradient in θ of its gradient in order to perform gradient descent. In consequence, the training process will be slow due to the need for the computation of the double gradient.

Denoising Score Matching. Fortunately, there is another way to perform score matching when p_t is the distribution of a random variable with gaussian noise added. Suppose that p is a density function and $q = p * g$ where g is another density function and $*$ is the convolution of two density functions.

Theorem 2 (Vincent (2011)). *Let $s : \mathbb{R}^d \to \mathbb{R}^d$ be a smooth function. Let X be a random variable with density p, ϵ an independent random variable with density g and $X_\epsilon = X + \epsilon$, whose density is $p_g = p * g$. Then,*

$$\mathbb{E}\Big[\|\nabla \log p_g(X_\epsilon) - s(X_\epsilon)\|^2\Big] = c + \mathbb{E}\big[\|\nabla \log g(\epsilon) - s(X_\epsilon)\|^2\big],$$

where c is a constant independent of s.

Applying this result to the setting of diffusion models, remember that p_t is the density of $e^{-\mu_t}X_0+\epsilon_t$ where $\epsilon_t \sim \prime, \sigma_{\sqcup}^{\in}$, hence in this case $g(x) = (2\pi\sigma_t^2)^{-d/2}e^{-|x|/2\sigma_t^2}$ and $\nabla \log g(x) = -x/\sigma_t^2$. We may now deduce the objective of denoising score matching to be

$$\arg\min_{\theta} \int_0^T w(t)\mathbb{E}\left[\left\| - \frac{\epsilon_t}{\sigma_t^2} - s_\theta(t, e^{-\mu_t}X_0 + \epsilon_t)\right\|^2\right] dt,$$

where X_0 is a random variable drawn uniformly from the dataset $\{x^{(1)}, \ldots, x^{(N)}\}$. At the cost of generating i.i.d. samples ξ_i from a standard Gaussian and τ_i uniform from $[0, T]$, we have access to (τ_i, X_i, ξ_i), where we may calculate in practice the following empirical version

$$\hat{\ell}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\left\|\frac{\xi_i}{\sigma_{\tau_i}} - s_\theta(\tau_i, e^{-\mu_t}X_i + \sigma_{\tau_i}\xi_i)\right\|^2\right].$$

2 Memorization effect

Generative models, particularly large-scale language models (LLMs) like OpenAI's GPT or Google's LaMDA but also diffusion models such as Stable Diffusion, have revolutionized artificial intelligence by enabling machines to produce human-like text, images, and even music. These models can generate creative and coherent content, answer complex questions, and engage in conversations. However, with their remarkable capabilities, comes a growing concern about the phenomenon known as "memorization." Memorization in the context of generative models refers to the models' ability to recall and regenerate specific information from their training data, which can raise issues related to data privacy, intellectual property, and the models' overall generalization capabilities.

In this section we delve into the memorization effect of generative models, its causes, potential consequences, and possible solutions. We'll also provide some empirical and theoretical evidence based on previous studies that targeted specifically upon diffusion models.

2.1 Types of Memorization

Memorization refers to the unintended capacity of generative models to produce exact replicas or near-verbatim text or data that appeared in their training dataset. This behavior contrasts with the intended purpose of these models, which is to generalize from the training data and generate novel outputs based on learned patterns rather than copying them.

In machine learning, a model's goal is to generalize from the data, meaning it should capture underlying patterns rather than rote learning or memorizing specific examples. Memorization occurs when the model "overfits" the training data, meaning it starts to store specific data points instead of understanding the data distribution. Generative models that memorize large chunks of their training data raise concerns when they unintentionally reproduce sensitive or proprietary information. In general, there are two types of memorization, see also Figure 2.

- **Verbatim Memorization**: This is the most straightforward form of memorization, where the model produces text or data that is identical or nearly identical to the training data.
- **Subtle Memorization**: In some cases, models may generate outputs that are not exact replicas but still contain recognizable patterns or phrases from the training data. This type is harder to detect but can still lead to privacy concerns.

2.2 Reasons of Memorization

The memorization effect in generative models is a consequence of the model's training process, its architecture, and the nature of the data it is trained on. Although, it has not been fully understood what are the actual reason causing memorization, we give some posiible factors that might contribute to memorization:

- **Model Size and Capacity**: The size of the model—often measured in billions or trillions of parameters—plays a critical role in memorization. Larger models with higher capacity can

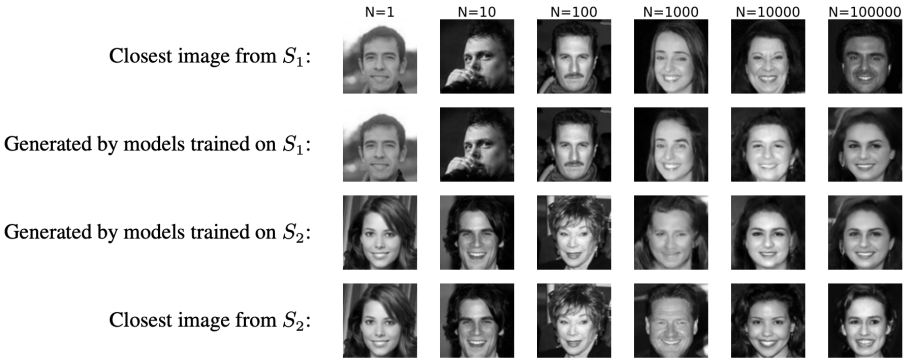

Figure 2: Memorization effect on diffusion models (S_1 and S_2) is highly correlated with the number of samples (N) in their training dataset. One finds verbatim memorization when N is significantly small, and subtle memorization when N is moderate. It only starts generalizing when N is significantly large enough.

store and recall more specific information from their training data. Research has shown that as model size increases, the likelihood of memorization also grows (Carlini et al., 2022; Zhang et al., 2021). Nevertheless, we cannot easily think larger capacity leads to more memorization because training data plays an important role. The effective capacity of networks cannot directly explain memorization and generalization (Arpit et al., 2017).

- **Long-Tail Data** (Feldman, 2020; Feldman and Zhang, 2020): Data that is rare or unique, often referred to as "long-tail data," can contribute to memorization. If a generative model encounters a rare phrase or sequence multiple times during training, it is more likely to memorize it and reproduce it during generation, especially if the sequence is distinctive or anomalous.
- **Overfitting**: Overfitting occurs when a model becomes too closely attuned to the training data, rather than learning to generalize from it. This overfitting can lead to the model memorizing specific examples, especially when trained on a finite dataset (Tirumala et al., 2022).

2.3 Ethical and Legal Implications of Memorization

Memorization by generative models presents ethical and legal challenges that must be addressed by both AI researchers and policymakers. These issues span data privacy, intellectual property, and the potential for misuse.

Data Privacy. One of the most significant concerns with memorization is the possibility that generative models may unintentionally expose sensitive or private information, such as:

Personally identifiable information (PII) Confidential business information Private communications Example: If a model trained on leaked datasets or improperly curated public data can regenerate exact passwords, credit card numbers, or personal emails, it could lead to significant privacy violations.

Intellectual Property. Models trained on copyrighted or proprietary content could regenerate parts of this content, raising questions about intellectual property infringement. This issue is particularly relevant in creative fields, such as writing, music composition, and visual arts. For example, a language model might memorize portions of a copyrighted book or song lyrics and reproduce them verbatim in response to certain prompts. This could lead to lawsuits over the unauthorized use of copyrighted material.

Bias and Misinformation. Memorization can also perpetuate biases and misinformation if the training data contains such problematic content. If a generative model memorizes and reproduces biased or false information, it can propagate harmful stereotypes or amplify misinformation. For example, a model trained on biased datasets might memorize and regenerate racist or sexist content. This problem

is particularly concerning in applications like news generation or customer support, where neutrality and fairness are paramount.

2.4 Theoretical studies of memorization on diffusion models

Understanding the causes of memorization and the importance of avoiding such effect. We show some theoretical study of memorization of diffusion models.

Recall the objective of denoising score matching for a given model s_θ parametrized by θ,

$$\ell(\theta) = \int_0^T w(t)\mathbb{E}\left[\left\| - \frac{\epsilon_t}{\sigma_t^2} - s_\theta(t, e^{-\mu_t}X_0 + \epsilon_t)\right\|^2\right],$$

which we may rewrite in the following form

$$\ell(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{t,X\sim p_t(\cdot|x^{(i)})}\left[\|s_\theta(t,X) - p_t(X|x^{(i)})\|^2\right], \tag{9}$$

where $p_t(\cdot|x)$ is the probability distribution of $\overrightarrow{X}_t$ in (1) given $\overrightarrow{X}_0 = x$, and t is a random variable drawn from $[0,T]$ whose density is given by $w(t)$. In fact, the $p_t(\cdot|x)$ is exactly the density of the Gaussian $\mathcal{N}(e^{-\mu_t}x, \sigma_t^2)$ whose score is equal to

$$s(t,z|x) = -\frac{z - \mu_t y}{\sigma_t^2}.$$

We may then show by direct computation (Li et al., 2024; Yi et al., 2023) that the solution of (9) is the following, for $(t,x) \in [0,T]\times\mathbb{R}^d$:

$$\hat{s}^N(t,x) = \frac{\sum_{k=1}^N s(t,x|x^{(k)})p_t(x|x^{(k)})}{\sum_{k=1}^N p_t(x|x^{(k)})}. \tag{10}$$

Theorem 3 (Simplified version of Theorem 4.3 in Li et al. (2024))**.** *Under some regularity assumption upon the dataset, the samples generated by (6) with the empirical optimal score function $\hat{s}^N$. Let q_t be the distribution of $\overleftarrow{X}_t$. Then this returns a simple Gaussian convolution with the empirical distribution in the form of $\frac{1}{N}\sum_{k=1}^N \mathcal{N}(\mu_t x^{(k)}, \sigma_t^2 I_d)$ whose density function we denote by $\hat{p}_t$, and it presents the following behavior:*

- ***With early stopping**: For any $\epsilon > 0$, set $T = \log\frac{d}{\epsilon}$ and $\delta = \frac{\epsilon^2}{d}$, we have*

$$TV(q_{T-\delta}, \hat{p}_\delta) \le \epsilon.$$

- ***Without early stopping**: By taking the limit $T\to\infty$ and $\delta = 0$, we have $q_\infty = \mathcal{U}(\{x^{(1)},\dots,x^{(N)}\})$ which is the uniform distribution of the dataset.*

In other words, Theorem 3 tells us that with the empirical optimal score $\hat{s}^N$ has a vebatim memorization effect if $T\to\infty$ and has a subtle memorization effect in finite time, which is illustrated in Figure 3.

3 Future Research

Understanding the memorization effect and the importance of preventing it from happening in practice, this opens up a lot of theoretical works to discover various way of avoiding memorizing. In this section, we list some of the possibilities for future research.

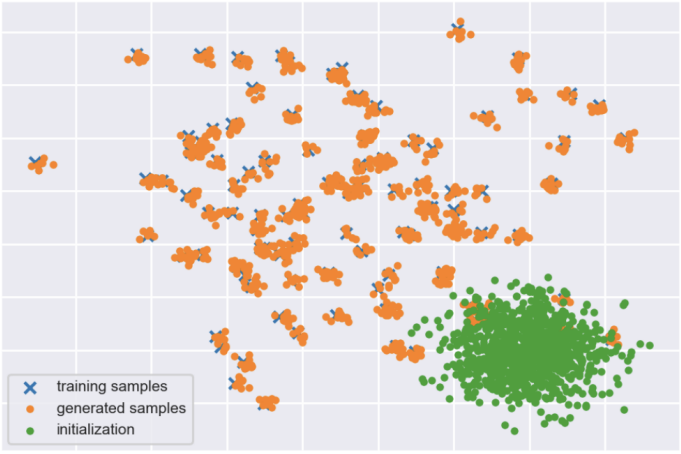

Figure 3: Samples generated by DDPM with empirical optimal score function $\hat{s}^N$ where the blue crosses are the training samples and orange dots are generated samples with early stopping time $\delta = 0.01$.

3.1 Implicit Regularization

Qiao et al. (2024) have shown that, the optimal solution of a regression problem using a 2-layer univariate neural network is regularized by the stepsize of the gradient descent.

To state their result formally, given a dataset $\{(x_i, y_i) \in \mathbb{R} \times \mathbb{R}\}_{1 \le i \le N}$ where there is a ground truth function f_0 and Gaussian noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ such that $f_0(x_i) + \epsilon_i = y_i$. The class of 2-layer univariate neural network is given by

$$\mathcal{F} = \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; f(x) = \sum_{i=1}^{k} w_i^{(2)} \phi\left(w_i^{(1)} x + b_i^{(1)} \right) + b^{(2)} \right\},$$

where the network consists of k hidden neurons and ϕ denotes the ReLU activation function. We update parameters of the model by the following gradient descent of stepsize η:

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t),$$

where $\mathcal{L}(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$ is the loss function and θ is the parameter of the function. With these setup, they have proven the following result

Theorem 4. *In the nonparametric regression problem with ground-truth function f_0, for an optimal model $f = f_\theta$ of gradient descent with step size η where $\mathcal{L}$ is twice differentiable at θ, assume that f is optimized, i.e, the empirical loss of f is smaller than f_0, then with probability $1 - \delta$, the function f satisfies*

$$\int_{-x_{\max}}^{x_{\max}} |f''(x)| g(x) dx \le \frac{1}{\eta} - \frac{1}{2} + \tilde{\mathcal{O}}(\sigma x_{\max}),$$

where $\tilde{\mathcal{O}}$ suppresses the logarithmic terms of N and $1/\delta$ and g is a weight function depending only on data $\{x_i\}_{1 \le i \le N}$.

In other words, they have shown that the optimal solution given by a 2-layer neural network trained by gradient descent should be smooth, which means that their second derivative should be upper bounded by the inverse of the stepsize. Therefore, the larger the stepsize the smoother the solution should be.

To apply their work, one may show that the empirical optimal score function given in (10) is not smooth when the number of data points are small, as shown in Figure 4. Therefore, it will be impossible for the model to learn the empirical optimal score with larger gradient descent stepsize. By doing so, the model will not memorize the training samples and thus training with larger stepsize implicitly circumvents memorization effect.

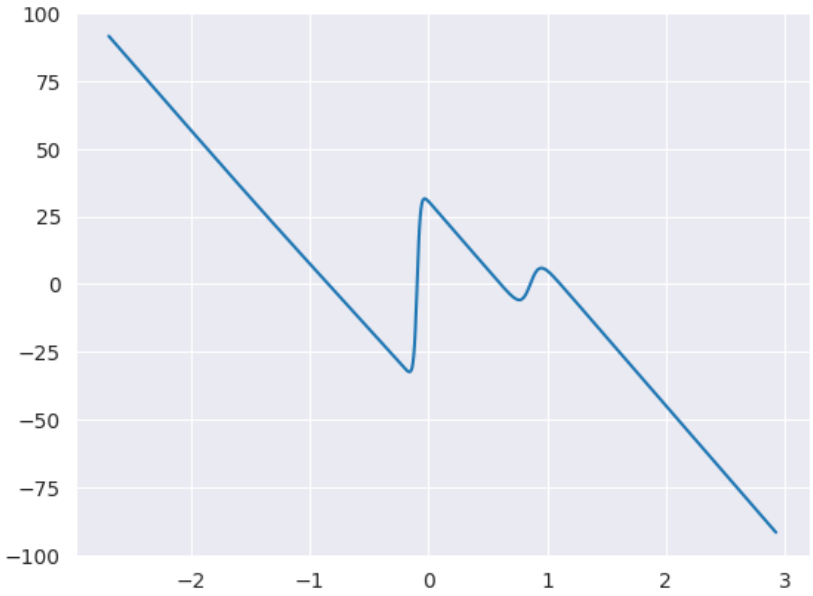

Figure 4: Empirical optimal score with 5 data points generated with Gaussian distribution.

3.2 Geometry-Adaptive Harmonic Basis

In order to understand the underlying geometric cause of the memorization effect, Kadkhodaie et al. (2023) have shown in their work that the diffusion models tend to learn the geometric-adaptive harmonic basis (GAHB) of the training samples. Suppose that the model f is bias free, then its input-output mapping is piecewise-linear. Thus, we can rewrite f in terms of Jacobian decomposition, i.e.,

$$f(t,x) = \nabla f(t,x)(t,x)^\top = \sum_k \lambda_k(t,x)\langle (t,x), e_k(t,x)\rangle e_k(t,x).$$

The denoiser can thus be interpreted as performing shrinkage with factors $\lambda_k(t,x)$ along axes of a basis specified by $e_k(t,x)$. They have shown that in the denoising score matching scenario (9), the optimal denoising error of an optimal model f_* at any fixed time t

$$\mathrm{MSE}(f_*) = \sigma_t^2 \mathbb{E}_x \sum_k \lambda_k(t,x),$$

where σ_t^2. Therefore, a small denoising error thus implies an approximately low-rank Jacobian (with many small eigenvalues) and thus an efficient approximation of x given y. In practice, the basis $e_k(t,x)$ is unknown. However, they've also shown empirically that diffusion models learn the basis vectors adapting to geometric boundary which defines the images, which is shown in Figure 5. On the contrary, when the border is perturbed by random noise, the model learns a suboptimal geometric structure as the decay of the eigenvalues is not as rapid. They have obeserved empirically that the better the model learns the inductive basis, the better the model generalizes. Therefore, establishing a theoretical result backing up their experiments may also help us understand the memorization effect from a geometric aspect.

4 Conclusion

The memorization effect of generative models presents both technical challenges and ethical dilemmas. As these models become increasingly powerful and widely used, it is crucial to develop strategies to mitigate memorization and ensure that generative AI aligns with societal values and legal frameworks. To this end, a fundamental understanding of memorization effect of diffusion models is inevitable and critical to address the issues.

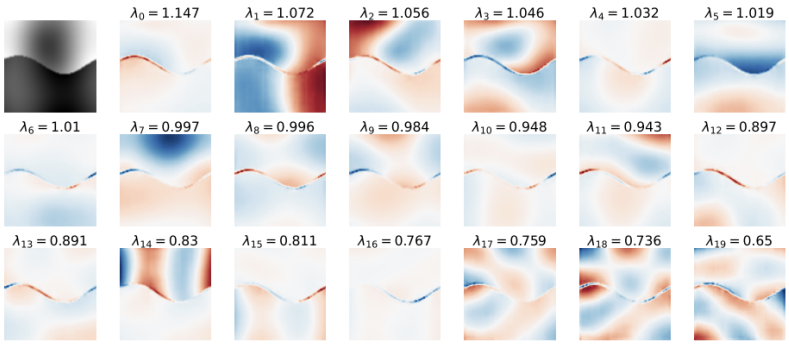

Figure 5: The model learns to adapt to the geometric boundary of the image.

The future of AI will depend on our ability to harness the creative potential of generative models while safeguarding against the unintended consequences of memorization.

Finally, I would like to express my sincere gratitude to Gérard and Pierre for their support and guidance throughout the end of my master. Their insights and feedback have been helpful in shaping the scope and directions of the study. Additionally, I thank LPSM for providing the necessary facilities and resources to conduct my research. Finally, I am thankful to Gérard, Pierre and Claire who will be the team of supervisors of my PhD along side Quentin and Romu who will be joining as co-advisor from Google.

References

J. M. L. Alcaraz and N. Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.

T. Amit, T. Shaharbany, E. Nachmani, and L. Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

B. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

N. Boffi and E. Vanden-Eijnden. Probability flow solution of the fokker–planck equation. arxiv 2022. *arXiv preprint arXiv:2206.04642*, 2022.

N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Z. Kadkhodaie, F. Guth, E. P. Simoncelli, and S. Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

S. Li, S. Chen, and Q. Li. A good score does not lead to a good generative model. *arXiv preprint arXiv:2401.04856*, 2024.

Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

D. Qiao, K. Zhang, E. Singh, D. Soudry, and Y.-X. Wang. Stable minima cannot overfit in univariate relu networks: Generalization by large step sizes, 2024. URL https://arxiv.org/abs/2406.06838.

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

N. Savinov, J. Chung, M. Binkowski, E. Elsen, and A. v. d. Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

M. Yi, J. Sun, and Z. Li. On the generalization of diffusion model. *arXiv preprint arXiv:2305.14712*, 2023.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.