PHASE-AWARE TRAINING SCHEDULE SIMPLIFIES LEARNING IN FLOW-BASED GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We analyze the training of a two-layer autoencoder used to parameterize a flow-based generative model for sampling from a high-dimensional Gaussian mixture. Building on the work of Cui et al. (2024), we find that the phase where the high-level features are learnt during training disappears as the dimension goes to infinity without an appropriate time schedule. We introduce a time dilation that solves this problem. This enables us to characterize the learnt velocity field, finding a first phase where the high-level feature (asymmetry between modes) is learnt and a second phase where the low-level feature (distribution of each mode) is learnt. We find that the autoencoder representing the velocity field learns to simplify by estimating only the parameters relevant to the feature for each phase. Turning to real data, we propose a method that, for a given feature, finds intervals of time where training improves accuracy the most on that feature, and we provide an experiment on MNIST validating this approach.

1 INTRODUCTION

In recent year, diffusion models have emerged as powerful techniques for learning to sample from high dimensional distributions Sohl-Dickstein et al. (2015); Song et al. (2021); Song & Ermon (2020); Ho et al. (2020), especially in the context of image generation and recently also for text Lou et al. (2024). The key idea lies in learning the gradient of the log density of corrupted data or *score function* from an optimization problem on samples and using the learnt *velocity* for generation. Despite the remarkable performance of these models, there remain several open questions, including understanding what makes a good noise-schedule, and how it is possible to overcome the curse of dimensionality.

We consider the problem of training a neural network to learn the velocity field to generate samples from a two-mode Gaussian Mixture (GM). This serves as a prototypical example to understand how diffusion models handle the learning of features at different scales. In the two-mode GM we have two well-separated scales: the macroscopic scale of the relative asymmetry between the modes, and the microscopic scale of the distribution of each mode.

This problem was previously considered by Cui et al. (2024) but their analysis only handles the *balanced* two-mode GM. On the other hand, when assuming access to the exact velocity field, Biroli et al. (2024) find an that the generative model for the two-mode Gaussian Mixture has a speciation time, defined as the time in generation, starting from noise, after which the mode where the sample will belong to is determined.

In this work, we show that if we analyze the estimation of the velocity field using the right time schedule so as to control where the speciation time happens, then we get accurate generation for the two-mode GM model. More precisely, our contributions are as follows.

- We give an asymptotic characterization of the learnt velocity field, finding a separation into two phases. We further show that having $\Theta_d(1)$ samples is enough to learn the velocity field.
- We show that the neural network representing the velocity field learns to simplify for each phase. In the first phase, it only concerns estimation of the high-level feature whereas in the second phase, it concerns estimation of the low-level feature. This sheds light on the

advantage of Diffusion Models over Denoising Autoencoders, since the sequential nature of Diffusion Models shown here allows them to decompose the complexity of the problem.

- For real data, this analysis suggests that training more at the times associated with a feature would improve accuracy on that feature. In fact, we propose a method that, given a feature, finds an interval of time where more traning improves accuracy on that feature the most. We further validate this on the MNIST dataset. We provide the code for the experiments here.

2 Related Works

Diffusions and flow-based generative models. Diffusion models Sohl-Dickstein et al. (2015); Song et al. (2021); Song & Ermon (2020); Ho et al. (2020) learn to invert the ODE/SDE that maps a given data distribution to Gaussian noise, for sampling. We refer the reader to Yang et al. (2024) for a review on methods and applications. Albergo et al. (2023) introduce the stochastic interpolant framework which allow for interpolation between two distributions, both deterministically and stochastically, in finite time.

Phase transitions of generative models in high dimensions. Several works analyze phase transitions in the dynamics of generative models. Raya & Ambrogioni (2023) find that diffusion models can exhibit symmetry breaking, where two phases are separated by a time where the potential governing the dynamics has an unstable fixed point. They give a full theoretical analysis for the data being two equiprobable point masses in $\mathbb{R}$, and also give a bound for the symmetry breaking time for the case where the data is a sum of finitely many point masses. Our setting generalizes the case of two equiprobable point masses in $\mathbb{R}$ to two Gaussians in $\mathbb{R}^d$ that are not necessarily equiprobable. Ambrogioni (2023) builds on Raya & Ambrogioni (2023) and shows several connections between equilibrium statistical mechanics and the phase transitions of diffusion models. Ambrogioni (2023) further conjectures that accurately sampling near times of "critical generative instability" affects the sample diversity. We give an explicit description of this critical times and verify this conjecture theoretically for sampling (see Proposition 1) and for learning (see Corollary 5) and empirically for learning (see Section 6). Li & Chen (2024) also formalize the study of critical windows taking the data to be a mixture of strongly log-concave densities. They give non-asymptotic bounds for the start and end times of these critical windows, which have a closed form expression for the mixtures of isotropic Gaussians case. In contrast, we provide sharp asymptotic characterizations for the phase transition times. Biroli & Mézard (2023) analyze the Curie-Weiss model and analyticalaly characterize the speciation time, defined as the time after which the mode that the sample will belong to is determined. Biroli et al. (2024) generalize the result and find an speciation time $t_s \sim \frac{1}{2}\log(\lambda)$ for an Ornstein-Uhlenbeck where λ is the largest eigenvalue of the covariance of the data, usually proportional to d. Montanari (2023) points out a similar phase transition when *learning* the velocity field to generate from a two-mode unbalanced Gaussian mixture, leading into problems for accurate estimation of the data. Montanari (2023) addresses this by using a different neural network to learn each mode. In the current work, we show that it is not needed to tailor the network for each mode if the right time schedule is used. It is worth noting that all these works are about sampling. We provide a result for sampling in Proposition 1. Building on this, we give results for learning (i.e. estimating the velocity field through a neural network) which is the main contribution of our paper.

Time-step complexity. Several results give convergence bounds detailing the required time-steps, score accuracy, and/or data distribution regularity to sample accurately. Benton et al. (2024) show that at most $O(d\log^2(1/\delta)/\epsilon^2)$ time steps are required to approximate a distribution corrupted with gaussian noise of variance δ to within ϵ^2 KL divergence. Chen et al. (2023) study probability flow ODE and obtain $O(\sqrt{d})$ convergence guarantees with an smoothness assumption. An underlying assumption in all these works is that the score or velocity field is learnt to certain accuracy. In the present work, we address this problem in the special case of a Gaussian mixture.

Sample complexity for Gaussian Mixtures. Cui et al. (2024) study the learning problem for the Gaussian mixture in high dimensions and demonstrate that $n = \Theta_d(1)$ samples are sufficient in the balanced case where the two modes have the same probability. This is done through statistical physics techniques of computing the partition function and using a sample symmetric ansatz. As we show, due to the speciation time at $d^{-1/2}$ which tends to zero as the dimension d grows, this analysis misses one phase of learning. Gatmiry et al. (2024) show that quasi-polynomial ($O(d^{\text{poly}(\log(\frac{d+k}{\epsilon}))})$)

sample and time complexity is enough for learning k-gaussian mixtures. The data distribution is more general than the one we consider, but on the other hand we give a $\Theta_d(1)$ sample and time complexity.

Statistical physics for analyzing neural networks. Bordelon et al. (2021); Canatar et al. (2021) use statistical physics methods to arrive at a notion of generalization which highlights the alignment of a model with the spectral bias in sample complexity of different features. In a broadly similar line, we will argue that that sample complexity in diffusion models is related to how well the schedule aligns with the different phases of the velocity field.

3 BACKGROUND

Data model and diffusion model. Consider the two-mode Gaussian Mixture Model (GMM)

$$\rho = p\mathcal{N}(\mu, \sigma^2 \mathrm{Id}_d) + (1-p)\mathcal{N}(-\mu, \sigma^2 \mathrm{Id}_d) \tag{1}$$

where $p \in (0,1)$ and $\mu \in \mathbb{R}^d$ such that $\|\mu\|^2 = d$ and $\sigma > 0$. A diffusion model for μ starts with samples from a simple distribution (say a Gaussian) and sequentially denoises them to get samples from the data.

More precisely, consider the stochastic interpolant

$$x_t = \alpha_t x_0 + \beta_t x_1 \tag{2}$$

where $x_0 \sim \mathcal{N}(0, \mathrm{Id}_d)$, $x_1 \sim \rho$, and $\alpha_t, \beta_t : [0,1] \to \mathbb{R}$, $\alpha_1 = 0 = \beta_0$, $\alpha_0 = 1 = \beta_1$. It is proven in Albergo et al. (2023) that if X_t solves the probability flow ODE

$$\dot{X}_t = b_t(X_t) \qquad \text{with} \qquad b_t(x) = \mathbb{E}[\dot{x}_t | x_t = x] \tag{3}$$

with $X_0 \sim \mathcal{N}(0, \mathrm{Id}_d)$, we then have for $t \in [0,1]$ that $X_t \overset{d}{=} x_t$ and hence $X_{t=1} \sim \rho$.

Since the data is coming from the GMM, the expression for the exact velocity field $b_t(x)$ from equation 3 can be computed exactly and it is given by combining equation 4 and equation 9 below. Our goal is to understand how well a neural network can estimate this velocity field through samples, in the large dimension $d \to \infty$ limit assuming low sample complexity for the data $n = \Theta_d(1)$.

Objective function. To fulfill our goal, we rewrite the velocity field as

$$b_t(x) = \left(\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t\right) f(x,t) + \frac{\dot{\alpha}_t}{\alpha_t} x, \tag{4}$$

where $f(x,t) = \mathbb{E}[x_1 | x_t = x]$ is the denoiser, which recovers the datapoint $a = x_1$ from a noisy version of it x_t. The denoiser is characterized as the minimizer of the loss (see Albergo et al. (2023))

$$\mathcal{R}[f] = \int_0^1 \mathbb{E}||f(x_t,t) - x_1||^2 dt. \tag{5}$$

In practice, however, we usually do not have access to the exact data distribution. So we assume we have a dataset $\mathcal{D} = \{x_1^\mu\}_{\mu=1}^n$ where $x_1^\mu \sim_{\text{iid}} \rho$. On the other hand, we have unlimited samples from $x_0 \sim \mathcal{N}(0, \mathrm{Id}_d)$. In practice, this means that to each data sample a_μ we can associate many noise samples $x_0^{\mu,\nu}$ with $\nu = 1, \cdots, k$. We then denote $x_t^{\mu,\nu} = \alpha_t x_0^{\mu,\nu} + \beta_t x_1^\mu$. At one step in our analysis, we will assume infinitely many noise samples associated to each data sample, so that we can take expectation with respect to the noise distribution. We also parameterize the denoiser through a different neural network for each t, which will be denoted as $f_{\theta_t}(x)$. This gives

$$\hat{\mathcal{R}}(\{\theta_t\}_{t\in[0,1]}) = \int_0^1 \sum_{\mu=1}^n \sum_{\nu=1}^k ||f_{\theta_t}(x_t^{\mu,\nu}) - x_1^\mu||^2 dt. \tag{6}$$

Finally, if we denote $\{\hat{\theta}_t\}_{t\in[0,1]}$ the minimizer of equation 6, we then define

$$\hat{b}_t(x) = \left(\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t\right) f_{\hat{\theta}_t}(x) + \frac{\dot{\alpha}_t}{\alpha_t} x, \tag{7}$$

and we can run the probability flow ODE

$$\dot{\hat{X}}_t = \hat{b}_t(\hat{X}_t)$$

with $\hat{X}_0 \sim \mathcal{N}(0, \mathrm{Id}_d)$.

Network architecture. We will analyze the case where the neural network parameterizing the denoiser function $f(x,t)$ consists of a two-layer Denoising Autoencoder with a trainable skip connection, motivated by the U-net Ronneberger et al. (2015)

$$f_{\theta_t}(x) = c_t x + u_t \tanh\left(\frac{w_t \cdot x}{\sqrt{d}} + b_t\right) \tag{8}$$

where $\theta_t = \{c_t, u_t, w_t, b_t\}$, $c_t, b_t \in \mathbb{R}$, and $u_t, w_t \in \mathbb{R}^d$. The structure of this Denoiser Autoencoder is highly motivated by the exact denoiser, which since the data is the GMM, can be computed exactly as follows (see Albergo et al. (2023), Appendix A)

$$\mathbb{E}[x_1 | x_t = x] = \frac{\beta_t \sigma^2}{\alpha_t^2 + \sigma^2 \beta_t^2} x + \frac{\alpha_t^2}{\alpha_t^2 + \sigma^2 \beta_t^2} \mu \tanh\left(\frac{\beta_t}{\alpha_t^2 + \sigma^2 \beta_t^2} \mu \cdot x + h\right) \tag{9}$$

where h is such that $e^h/(e^h + e^{-h}) = p$.

We will consider the loss with regulatization for w_t and u_t. Since for each t we assume a different neural network, we can write the loss as

$$\hat{\mathcal{R}}_t(\theta_t) = \sum_{\mu=1}^{n} \sum_{\nu=1}^{k} ||f_{\theta_t}(x_t^{\mu,\nu}) - x_1^{\mu}||^2 + \frac{\lambda}{2}||u_t||^2 + \frac{\ell}{2}||w_t||^2. \tag{10}$$

We note that Cui et al. (2024) consider the special case of tied weights $u_t = w_t$ and $b_t = 0$. This is enough to capture the balanced GMM (i.e. $p = 1/2$) but fails at learning to sample from the GMM for $p \neq 1/2$. This follows since x_0 has an even distribution and their choice of tied weights and no bias yields an odd velocity field which results in an even distribution for x_t. If the bias is added and the weights are untied, the analysis of Cui et al. (2024) will still not able to learn to sample for $p \neq 1/2$. This is because the gradient for w_t vanishes as $d \to \infty$ unless special care is given to the small times where the phase transition happens, as will be explained next.

Separation into phases. The analysis of Biroli et al. (2024) assumes access to the exact velocity field, and shows that the generative model from equation 3 with $\alpha_t = \sqrt{1-t^2}$ and $\beta_t = t$ undergo a phase transition at time $t_s = 1/\sqrt{d}$. They call this the speciation transition, and it is defined as the time in the generation process after which the mode that the sample will belong to at the end of the process is determined. It is straightforward to check that their analysis shows that the speciation transition would also be $t_s = 1/\sqrt{d}$ if we instead consider $\alpha_t = 1 - t$ and $\beta_t = t$ which are the choices that we will use for our analysis. We will not prove this since this result is only to motivate our work and will not be used in the proofs.

Having $t_s = 1/\sqrt{d}$ as speciation time means that in the asymptotic limit $d \to \infty$, the t_s goes to zero and the possibility of learning the relative asymmetry between the modes is lost. This is the essence of why the analysis of Cui et al. (2024) can not learn p for $p \neq 1/2$.

We will dilate time so as to make the speciation time t_s not disappear as $d \to \infty$. More precisely, consider the time dilation $\tau : [0,2] \to [0,1]$

$$\tau(t) = \begin{cases} \frac{\kappa t}{\sqrt{d}} & \text{if } t \in [0,1] \\ \frac{\kappa}{\sqrt{d}} + \left(1 - \frac{\kappa}{\sqrt{d}}\right)(t-1) & \text{if } t \in [1,2]. \end{cases} \tag{11}$$

We note this fulfills $\tau(0) = 0, \tau(1) = \kappa/\sqrt{d}$, and $\tau(2) = 1$. Running the generative model with this time dilation, we find that at generation time there is a separation into two phases: the first phase with $t \in [0,1]$ where the high-level feature (the asymmetry between the modes) is estimated, and the second phase with $t \in [1,2]$ where the low-level feature (the distribution of each mode) is estimated. Further, the velocity fields are independent of d.

Proposition 1. *Let X_t be the solution to the probability flow ODE from equation 3 with $\alpha_t = 1 - \tau_t$ and $\beta_t = \tau_t$ where τ_t is defined in equation 11. Then*

$$X_t - \frac{\mu \cdot X_t}{d}\mu \sim \mathcal{N}\left(0, \sigma_t^2 Id_{d-1}\right).$$

where σ_t fulfills the following

- ***First phase**: For $t \in [0, 1]$, we have*

$$\lim_{d\to\infty} \sigma_t = 1.$$

  In addition, $\nu_t = \lim_{d\to\infty} \frac{\mu \cdot X_t}{\sqrt{d}}$ fulfills

$$\nu_1 \sim p\mathcal{N}(\kappa, 1) + (1-p)\mathcal{N}(-\kappa, 1).$$

- ***Second phase**: We have*

$$\lim_{d\to\infty} \sigma_2 = \sigma.$$

  In addition, $M_t = \lim_{d\to\infty} \frac{\mu \cdot X_t}{d}$ fulfills

$$M_2 \sim p^\kappa \delta_1 + (1 - p^\kappa)\delta_{-1}$$

  where p^κ is such that $\lim_{\kappa\to\infty} p^\kappa = p$

See Appendix A for the proof. Also see Appendix E for a generalization of the time dilation formula in equation 11 for a GM with more than two modes.

It is straightforward to see that $\operatorname{sgn}(\mu \cdot X_t/d)$ stays constant during the second phase with high probability (to be precise, with probability going to 1 as $d \to \infty$ and $\kappa \to \infty$.) This means that in the first phase the high-level feature (the asymmetry between the modes) is learnt. This can be also seen from the fact that p appears in the velocity field of the first phase through h, but does not appear in the velocity field of the second phase. On the other hand, the variance of each mode σ^2 only appears in the second phase.

Rephrasing our previous discussion, the analysis of Cui et al. (2024) can not capture the learning of the parameters for $p \neq 1/2$ because the first phase (where this parameter is learnt) disappears as $d \to \infty$. Using the time dilation from equation 11, the first phase does not disappear. *In the present work, we show that this time dilation allows us to learn to sample from the GMM for $p \in [0, 1]$.*

We will show this in two steps. First, in Section 4 we show that we can characterize the learnt parameters of the velocity field in terms of a few projections, called the overlaps. Then, in Section 5 using the characterization of parameters in terms of overlaps, we show that using the learnt velocity field for generation recovers both the high-level feature (given by the relative asymmetry) and the low-level feature (given by the distribution of each mode.)

4 LEARNING

In this and the next section, we will assume we have access to n data samples and ask how well we can generate the target distribution using a learnt denoiser as $d \to \infty$. More precisely, we let $\hat{\theta}_t$ be the minimizer of the loss from equation 10. Then we parameterize an estimate of the denoiser $\hat{f}_{\hat{\theta}_t}$ as in equation 1, use this to get an estimated velocity field $\hat{b}$ as in equation 7, and run an ODE with this velocity field, whose solution we denote by $\hat{X}_t$. In this section, we will characterize $\hat{\theta}_t$ in the $d \to \infty$ limit. In the next section, we show that the result of running an ODE with the estimated denoiser gives a distribution that approaches the target distribution when we take first $d \to \infty$ and then $n \to \infty$ meaning that we have low-sample complexity, $n = \Theta_d(1)$.

We make the analysis concrete by considering $\alpha_t = 1 - \tau_t$, $\beta_t = \tau_t$. As mentioned in Section 3, the analysis will use the time dilation from equation 11. We will first analyze the times $t \in [0, 1]$ from the first phase and then times $t \in [1, 2]$ from the second phase.

4.1 First Phase

The interpolant in the first phase reads

$$x_t^\mu = \left(1 - \frac{\kappa t}{\sqrt{d}}\right) x_0^\mu + \frac{\kappa t}{\sqrt{d}} x_1^\mu$$

where $t \in [0, 1]$. We introduce overlaps in terms of which we will characterize the loss from equation 10 (note that all of these are functions of t but we drop the dependence for notational simplicity.)

$$p_\eta^\mu = \frac{z^\mu \cdot w}{d} \quad \omega = \frac{\mu \cdot w}{d} \quad r = \frac{\|w\|^2}{d} \quad q_\xi^\mu = \frac{x_0^\mu \cdot u}{d} \quad q_\eta^\mu = \frac{z^\mu \cdot u}{d} \quad m = \frac{\mu \cdot u}{d} \quad q = \frac{\|u\|^2}{d} \tag{12}$$

The following result gives equations for the overlaps in the asymptotic $d \to \infty$ limit which can be solved numerically.

This allows us to explicitly give the values of the overlaps as $n \to \infty$, which we use in Section 5 to argue that running a generative model with learnt parameters leads to accurate generation.

Result 1 (Sharp Characterization of Parameters in First Phase)**.** *For any $t \in [0, 1]$, in the $d \to \infty$ limit, the parameters minimizing the loss from equation 10 satisfy the following set of equations*

$$q_\eta = \frac{\sigma \overline{\phi}}{\lambda + n\overline{\phi^2}}$$

$$m = \frac{n\overline{\phi s}}{\lambda + n\overline{\phi^2}}$$

$$c = q_\xi = p_\eta = 0$$

$$q = m^2 + nq_\eta^2$$

$$r = \omega^2$$

$$(\lambda + n\overline{\phi^2})(\sigma(\overline{\phi'})(\overline{\phi}) + n(\overline{\phi' s})(\overline{\phi s})) = (n^2\overline{\phi s}^2 + n\sigma^2\overline{\phi}^2)(\overline{\phi'\phi})$$

$$\hat{r}(\lambda + n\overline{\phi^2})^2 = -n((\lambda + n\overline{\phi^2})(\sigma(\overline{\phi''})(\overline{\phi}) + n(\overline{\phi'' s})(\overline{\phi s})) - (n^2\overline{\phi s}^2 + n\sigma^2\overline{\phi}^2)(\overline{\phi\phi'})')$$

$$\omega(\ell + \hat{r})(\lambda + n\overline{\phi^2})^2 = (n\kappa t)((\lambda + n\overline{\phi^2})(\sigma(\overline{\phi' s})(\overline{\phi}) + n(\overline{\phi'})(\overline{\phi s})) - (n^2\overline{\phi s}^2 + n\sigma^2\overline{\phi}^2)(\overline{\phi' \phi s}))$$

here and in what follows we denote

$$\overline{y} = \frac{1}{nk}\sum_{\mu=1}^{n}\sum_{\nu=1}^{k} \mathbb{E}_{z^{\mu,\nu}}[y^{\mu,\nu}] = \overline{p}\mathbb{E}_{z^{\mu,\nu}}[y^{\mu,\nu}|s^\mu = 1] + (1 - \overline{p})\mathbb{E}_{z^{\mu,\nu}}[y^{\mu,\nu}|s^\mu = -1].$$

See Appendix B.1 for a heuristic derivation of this result, at the rigor level of theoretical physics. We also show in Appendix B.1.1 that using the equations from Result 1 and taking $n \to \infty$ gives very simple explicit equations for the overlaps

Corollary 1 (Parameters given infinite samples)**.** *For any $t \in [0, 1]$, taking $d \to \infty$ and then $n \to \infty$ gives the following overlaps*

$$c = q_\xi = q_\eta = p_\eta = 0$$
$$m = 1$$
$$\omega = \kappa t$$
$$\tanh(b) = 2(p - \tfrac{1}{2})$$

Note that the overlaps in the $n \to \infty$ limit do not contain any information about σ^2. This means that the estimation of σ happens completely in the second phase. We now turn to getting formulas for the Mean Squared Error. Define the scaled train and test MSE of the denoiser as

$$\text{mse}_{\text{train}} = \frac{1}{dnk}\sum_{\mu=1}^{n}\sum_{\nu=1}^{k} ||f_{\theta_t}(x_t^{\mu,\nu}) - x_1^\mu||^2,$$

$$\text{mse}_{\text{test}} = \frac{1}{d}\mathbb{E}\left[\|f_{\hat{\theta}_t}(x_t) - x_1\|^2\right].$$

Using the above results we characterize the MSE as follows

Corollary 2 (Mean Squared Error). *In the limit of $d \to \infty$,*

$$mse_{train} = 1 + \sigma^2 + c^2 + q\overline{\phi^2} - 2(\overline{s\phi}m + (\sigma q_\eta - cq_\xi)\overline{\phi})$$
$$mse_{test} = 1 + \sigma^2 + c^2 + q\overline{\phi^2} - 2\overline{s\phi}m$$

These reduce to

$$mse_{train} = 1 + \sigma^2 - \frac{(2\lambda + \overline{\phi^2})(n^2\overline{\phi s} + n\sigma\overline{\phi}^2)}{(\lambda + n\overline{\phi^2})^2}$$
$$mse_{test} = 1 + \sigma^2 - \frac{(2\lambda + \overline{\phi^2})n^2\overline{\phi s}}{(\lambda + n\overline{\phi^2})^2}$$

For $n \to \infty$, we get

$$mse_{train} = mse_{test} = \sigma^2 + (1 - \overline{\phi s}).$$

4.2 Second Phase

We now consider times $t \in [1, 2]$ which means we have

$$x_t^\mu = (2 - t)\left(1 - \frac{\kappa}{\sqrt{d}}\right)x_0^\mu + \left(\frac{\kappa}{\sqrt{d}} + \left(1 - \frac{\kappa}{\sqrt{d}}\right)(t - 1)\right)x_1^\mu.$$

Using similar definitions of overlaps as for the first phase (see Appendix B.2 for exact definitions), we find closed-form equations for the overlaps in the asymptotic $d \to \infty$ limit, and again find the limit as $n \to \infty$ for the overlaps. See Appendix B.2 for a heuristic derivation of this result

Result 2 (Sharp Characterization of Parameters in Second Phase). *For any $t \in [1, 2]$, in the $d \to \infty$ limit, the parameters minimizing the loss from equation 10 satisfy the following equations*

$$q_\xi = \frac{c(1 - \tau)}{\lambda + n}$$
$$q_\eta = \frac{\sigma(1 - c\tau)}{\lambda + n}$$
$$m = \frac{n(1 - c\tau)}{\lambda + n}$$
$$q = m^2 + nq_\xi^2 + n\sigma^2 q_\eta^2$$
$$c = \frac{\tau\left((1 + \sigma^2)(\lambda + n) - (\sigma + n)\right)}{(\lambda + n)((1 - \tau^2) + (1 + \sigma^2)\tau^2) + ((1 - \tau)^2 - \tau^2(\sigma + n))}$$

where $\tau = t - 1$.

Corollary 3 (Parameters given inifite samples). *For any $t \in [1, 2]$, taking $d \to \infty$ and then $n \to \infty$ gives the following overlaps*

$$c = \frac{\tau\sigma^2}{1 + (\sigma^2 - 1)\tau^2}$$
$$q_\xi = q_\eta = 0$$
$$m = 1 - c\tau$$

where $\tau = t - 1$.

In an opposite way to what happens in the second phase, we see that the parameter p does not appear in the overlaps whereas now σ does. This combined with the behavior in the first phase, shows that there is a separation into phases that can be learned by the generative model.

Corollary 4 (Mean Squared Error). *In the limit of $d \to \infty$, we have*

$$mse_{train} = (1 + \sigma^2)(1 - c\tau)^2 + c^2(1 - \tau)^2 + q - 2(1 - c\tau)(\sigma q_\eta + m) + 2c(1 - \tau)q_\xi$$
$$mse_{test} = (1 + \sigma^2)(1 - c\tau)^2 + c^2(1 - \tau)^2 + q - 2(1 - c\tau)m$$

For $n \to \infty$, we get

$$mse_{train} = mse_{test} = (1 + \sigma^2)(1 - c\tau)^2 + c^2(1 - \tau)^2$$

5 GENERATION

We show that using the probability flow ODE with the learnt denoiser recovers both paramters p and σ^2 from the target distribution. Let X_t be the solution to the ODE from equation 3 using the exact denoiser from equation 9, and let $\hat{X}_t$ be the solution using the learnt denoiser whose parameters we characterized in Section 4 (see the beginning of Section 4 for the exact definition of $\hat{X}_t$.) Assume X_t and $\hat{X}_t$ have a shared initial condition $X_{t=0} = \hat{X}_{t=0} \sim \mathcal{N}(0, \mathrm{Id}_d)$. Then $X_t - \hat{X}_t$ fulfills an ODE with initial condition 0 whose velocity field is in the span of u_t and μ.

We get from Result 1 that in the first phase $q = m^2 + nq_\eta^2$. This can be explicitly stated as

$$\lim_{d\to\infty} \frac{\|u\|^2}{d} = \lim_{d\to\infty} \left(\frac{\mu \cdot u}{d}\right)^2 + \left(\frac{\eta \cdot u}{d}\right)^2$$

where $\eta = \sigma \sum_{\mu=1}^{n} z^\mu$. This means that u_t is asymptotically contained in $\mathrm{span}(\mu, \eta)$, in the sense that the projection to the complement of $\mathrm{span}(\mu, \eta)$ has asymptotically vanishing norm, for $t \in [0, 1]$. Similarly, from Result 2, we get $q = m^2 + nq_\xi^2 + nq_\eta^2$, which means that u_t is asymptotically contained in $\mathrm{span}(\mu, \eta, \xi)$ for $t \in [1, 2]$ where $\xi = \sum_\mu s^\mu x_0^\mu$. This means that to show that X_t is close to $\hat{X}_t$, it suffices to bound the projections of $X_t - \hat{X}_t$ onto μ, η, and ξ. In fact, we have the following result (see Appendix C)

Result 3. *Let X_t be the solution of the probability flow ODE from equation 3 using the exact denoiser from equation 9. Let $\hat{X}_t$ be the solution using the learnt denoiser. Assume $X_{t=0} = \hat{X}_{t=0} \sim \mathcal{N}(0, Id_d)$. Then for $w \in span(\mu, \eta, \xi)$, with $\|w\|_2 = 1$, we have*

$$\lim_{d\to\infty} \frac{w \cdot (X_2 - \hat{X}_2)}{\sqrt{d}} = O\left(\frac{1}{n}\right).$$

For $w \in span(\mu, \eta, \xi)^\perp$, with $\|w\|_2 = 1$, we have

$$\lim_{d\to\infty} \frac{w \cdot (X_2 - \hat{X}_2)}{\sqrt{d}} = 0.$$

Corollary 5 (Parameters p and σ^2 are estimated correctly)**.** *Let $\hat{X}_t$ be the solution of the probability flow ODE from equation 3 using the learnt denoiser, starting with $\hat{X}_0 \sim \mathcal{N}(0, Id_d)$. We have*

$$\lim_{\kappa\to\infty} \lim_{n\to\infty} \lim_{d\to\infty} \frac{\mu \cdot \hat{X}_2}{d} \sim p\delta_1 + (1-p)\delta_{-1}.$$

For $w \perp \mu$, with $\|w\| = 1$, we have

$$\lim_{n\to\infty} \lim_{d\to\infty} \frac{w \cdot \hat{X}_2}{\sqrt{d}} \sim \mathcal{N}(0, \sigma^2).$$

We conclude that thanks to our time dilation, the distribution generated using the learnt denoiser captures both p and σ^2.

6 EXPERIMENTS

6.1 VERIFICATION OF HIGH-LEVEL FEATURE BEING CAPTURED

To show that the difference between the time dilated interpolant and the non time dilated one appears in practice, we first run a simple experiment. We run Gradient Descent with the Adam optimizer Diederik (2014) to learn the parameters w_t, c_t, u_t, b_t in equation 1 both for $\alpha_t = 1 - t, \beta_t = t$ and the dilated version $\alpha_t = 1 - \tau_t, \beta_t = \tau_t$. The results are in Figure 1. It is clear that the non dilated interpolant is not able to estimate the relative asymmetry correctly whereas the dilated interpolant is able to.

The code for this experiment is available here.

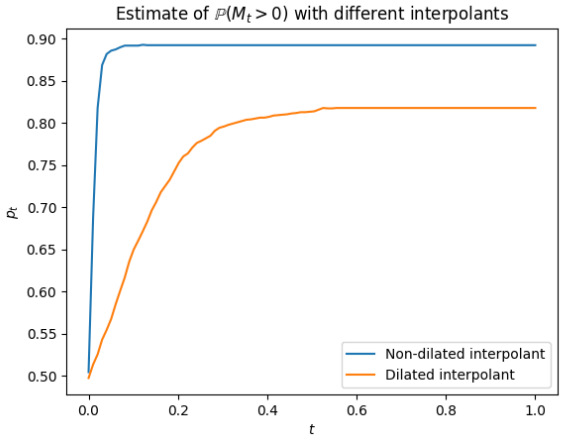

Figure 1: We learn the parameters from for different choices of interpolant. In all experiments, we take 100 discretization points, train for 5000 epochs, $n = 128$, $d = 5000$ with $p = .8$. After having learnt the parameters, we run the probability flow ODE with the learnt parameters for $K = 2000$ realizations $(X_t^j)_{2000}^{j=1}$ and then compute $p_t = \frac{1}{K}\sum_{j=1}^{K} \mathbb{I}\left[\sum_{i=1}^{d}(X_t^{i,j}) > 0\right] = \frac{1}{K}\sum_{j=1}^{K} \mathbb{I}\left[M_t^j > 0\right]$, which is an estimate of $\mathbb{P}(M_t > 0) = p$ where $M_t = \mu \cdot X_t/d$. We learn the parameters for with $\alpha_t = 1 - t, \beta_t = t$ which gives the non-dilated interpolant in blue. We predict the speciation to happen approximately at $\frac{1}{\sqrt{5000}} \approx .014$. Practice agrees with this, since we see most of the speciation happening at the first two ODE steps. We also learn the parameters for with $\alpha_t = 1 - \tau_t, \beta_t = \tau_t, \kappa = 4$. This gives the dilated interpolant plot in orange. We see the dilated interpolant estimates $p = .8$ much better than the non-dilated one.

6.2 Training a given feature on real data

Recall that in the background we mentioned that the analysis of Biroli et al. (2024) shows that taking $\alpha_t = 1 - t$ and $\beta_t = t$ without any time-dilation gives an speciation time $t_s = 1/\sqrt{d}$. This then means that the relative asymmetry between the modes (given by p) can not be captured as $d \to \infty$. Our analysis then shows that if we dilate time by streching the interval $[0, \kappa/\sqrt{d}]$ to $[0, 1]$ and the interval $[\kappa/\sqrt{d}, 1]$ to $[1, 2]$, then we get accurate estimation of p.

When training diffusion models in practice, we first sample a batch of times $t_1, \cdots, t_k$ uniformly. We then draw $x_0^\mu \sim \mathcal{N}(0, \mathrm{Id}_d)$, x_1^μ from our data distribution, and form a noisy sample $x_{t^\mu}^\mu = (1 - t^\mu)x_0^\mu + t^\mu x_1^\mu$ for $\mu = 1, \cdots, k$. We finally train on the loss

$$\hat{\mathcal{R}}(\theta) = \sum_{\mu=1}^{k} ||f_\theta(x_{t^\mu}^\mu, t^\mu) - x_1^\mu||^2. \tag{13}$$

where we took time as a parameter of the network as it is usually done in practice, as opposed to having a separate network for each time t.

The insight of our analysis is that instead of taking the batch of times uniformly, we can sample more times near the phase transition associated to a given feature, and in this way improve accuracy on that feature.

For a given feature, we can find the times where that feature is learnt using the U-Turn method (Sclocchi et al. (2024), Biroli et al. (2024)). Consider a dataset where each sample corresponds to exactly one of finitely many classes. Examples of this are samples of the GMM which correspond to one of two modes, or samples of MNIST which correspond to one of ten digits. The U-Turn then consists of starting with a sample from the data, run a backward diffusion model from time $t = 1$

to $t = t_0$, which noises the sample, and then run the forward diffusion model from time $t = t_0$ to $t = 1$ with noise independent from the backward run.

We are then interested in the probability that the sample before the backward and forward passes belongs to the same class as the sample after them. For $t_0 \approx 1$, this probability is close to 1. For $t_0 \approx 0$, this probability is close to the underlying probability of the diffusion model generating a sample of the given class. By running this for different t_0, we can find at what times it is decided to what class the samples belong to. Having found those times, our goal is to have a model that generates samples for each class according to the probabilities that they appear in the dataset. We can then improve the accuracy of the model on this by training on these times.

As a simple example, we train a U-Net (see Appendix D for details) to parameterize the Variance Preserving SDE from Song et al. (2021) to generate either the 0 or 1 digits from MNIST. The dataset we train on consists of 20% 1 digits and 80% 0 digits. We then measure how well is this model in generating samples that represent this asymmetry. The model is trained on approximately 7400 samples for 9 epochs, by sampling times in $[0, 1]$ uniformly as described in the beginning of this section. We then generate 18500 new samples running this model using 1000 discretization steps. [1] Among the 18500 generated samples, 88.2% are digits 0. (For determining this, we used a discriminator with 99.2% accuracy on MNIST, see Appendix D for details.)

We then test our proposed method. First, we determine at what time the digit that the sample represents is decided. We do this with the U-Turn method described above. Note that to do this, we use the model that we already trained. The results are in Figure 2. We find that the times important for deciding the digit are early in the generation for $t \in [0.2, 0.6]$ and mostly concentrated on $t \in [0.3, 0.5]$.

We now train from scratch a model on 7400 samples for 9 epochs as before, except that we do not sample the times uniformly. We instead sample times with probability $1/2$ uniformly in the interval $[0.3, 0.5]$ and with probability $1/2$ uniformly outside that interval. We then generate 18500 new samples with this new model using 1000 discretization steps, and find that 81.0% are 0s. We similarly consider sampling times with probability $1/2$ uniformly in the interval $[0.2, 0.6]$ and with probability $1/2$ outside that interval, generate samples, and find that 81.1% are 0s. This validates our hypothesis in the simple case of MNIST.

Although our theoretical analysis is for the probability flow ODE on the GMM data distribution, this example on MNIST shows that the ideas developed here can be useful to the SDE generative models used in practice for real data.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL https://arxiv.org/abs/2303.08797.

Luca Ambrogioni. The statistical thermodynamics of generative diffusion models. *arXiv preprint arXiv:2310.17467*, 2023.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization, 2024. URL https://arxiv.org/abs/2308.03686.

Giulio Biroli and Marc Mézard. Generative diffusion in very large dimensions. *Journal of Statistical Mechanics: Theory and Experiment*, 2023(9):093402, September 2023. ISSN 1742-5468. doi: 10.1088/1742-5468/acf8ba. URL http://dx.doi.org/10.1088/1742-5468/acf8ba.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models, 2024. URL https://arxiv.org/abs/2402.18491.

[1] This amount of discretization steps is much larger than what is needed for MNIST, and we do it this way to make sure that the error is not coming from the integration of the SDE but from the training alone.

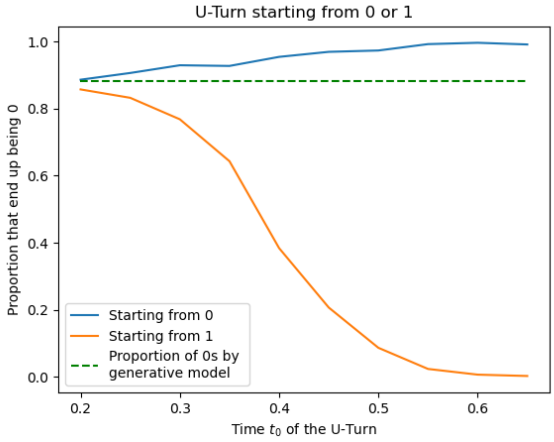

Figure 2: For $t_0 \in [0.2, 0.65]$, we plot the proportion of 0s that we get by doing the U-Turn at tiem t_0 starting from either 0 or 1 at time $t = 1$. On dashed green, we plot $y = .882$ which is the estimated proportion of 0s that the diffusion model generates starting from noise.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks, 2021. URL https://arxiv.org/abs/2002.02561.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1), May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23103-1. URL http://dx.doi.org/10.1038/s41467-021-23103-1.

Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast, 2023. URL https://arxiv.org/abs/2305.11798.

Hugo Cui, Florent Krzakala, Eric Vanden-Eijnden, and Lenka Zdeborová. Analysis of learning a flow-based generative model from limited sample complexity, 2024. URL https://arxiv.org/abs/2310.03575.

P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014.

Khashayar Gatmiry, Jonathan Kelner, and Holden Lee. Learning mixtures of gaussians using diffusion models, 2024. URL https://arxiv.org/abs/2404.18869.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Farley Knight. MNIST Digit Classification Model. https://huggingface.co/farleyknight/mnist-digit-classification-2022-09-04, 2022. Accessed: [September 28, 2024].

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models. *arXiv preprint arXiv:2403.01633*, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024. URL https://arxiv.org/abs/2310.16834.

Andrea Montanari. Sampling, diffusions, and stochastic localization, 2023. URL https://arxiv.org/abs/2305.10690.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models, 2023. URL https://arxiv.org/abs/2305.19693.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data, 2024. URL https://arxiv.org/abs/2402.16991.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. URL https://arxiv.org/abs/1907.05600.

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.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024. URL https://arxiv.org/abs/2209.00796.