ANALYSIS OF LEARNING A FLOW-BASED GENERATIVE MODEL FROM LIMITED SAMPLE COMPLEXITY

Hugo Cui
Statistical Physics of Computation Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Florent Krzakala
Information Learning and Physics Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Eric Vanden-Eijnden
Courant Institute of Mathematical Science
New York University (NYU)
New York, USA

Lenka Zdeborová
Statistical Physics of Computation Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

ABSTRACT

We study the problem of training a flow-based generative model, parametrized by a two-layer autoencoder, to sample from a high-dimensional Gaussian mixture. We provide a sharp end-to-end analysis of the problem. First, we provide a tight closed-form characterization of the learnt velocity field, when parametrized by a shallow denoising auto-encoder trained on a finite number n of samples from the target distribution. Building on this analysis, we provide a sharp description of the corresponding generative flow, which pushes the base Gaussian density forward to an approximation of the target density. In particular, we provide closed-form formulae for the distance between the means of the generated mixture and the mean of the target mixture, which we show decays as $\Theta_n(1/n)$. Finally, this rate is shown to be in fact Bayes-optimal.

Flow and diffusion-based generative models have introduced a shift in paradigm for density estimation and sampling problems, leading to state-of-the art algorithms e.g. in image generation (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022). Instrumental in these advances was the realization that the sampling problem could be recast as a transport process from a simple –typically Gaussian– base distribution to the target density. Furthermore, the velocity field governing the flow can be characterized as the minimizer of a quadratic loss function, which can be estimated from data by (a) approximating the loss by its empirical estimate using available training data and (b) parametrizing the velocity field using a denoiser neural network. These ideas have been fruitfully implemented as part of a number of frameworks, including score-based diffusion models (Song & Ermon, 2019; Song et al., 2020; Karras et al., 2022; Ho et al., 2020), and stochastic interpolation (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Lipman et al., 2022; Liu et al., 2022). A tight analytical understanding of the learning of generative models from limited data, and the resulting generative process, is however still largely missing. This constitutes the research question addressed in the present manuscript.

A line of recent analytical works (Benton et al., 2023; Chen et al., 2022; 2023a;c;d; Wibisono & Yang, 2022; Lee et al., 2022; 2023; Li et al., 2023; De Bortoli et al., 2021; De Bortoli, 2022; Pidstrigach, 2022; Block et al., 2020) have mainly focused on the study of the transport problem, and provide rigorous convergence guarantees, taking as a starting point the assumption of an L^2-accurate estimate of the velocity or score. They hence bypass the investigation of the learning problem –and in particular the question of ascertaining the sample complexity needed to obtain such an accurate estimate. More importantly, the study of the effect of learning from a *limited* sample complexity (and thus e.g. of possible network overfitting and memorization) on the generated density, is furthermore left unaddressed. On the other hand, very recent works (Cui & Zdeborová, 2023; Shah et al., 2023) have characterized the learning of Denoising Auto-Encoders (DAEs) (Vincent et al., 2010; Vincent, 2011) in high dimensions on Gaussian mixture densities. Neither work however studies the consequences on the generative process. Bridging that gap, recent works have offered a *joint* analysis of the learning and generative processes. Oko et al. (2023); Chen et al. (2023b); Yuan

et al. (2023) derive rigorous bounds at finite sample complexity, under the assumption of data with a *low-dimensional* structure. Closer to our manuscript, a concurrent work (Mei & Wu, 2023) bounds the Kullback-Leibler distance between the generated and target densities, when parametrizing the flow using a ResNet, for high-dimensional graphical models. On the other hand, these bounds do not go to zero as the sample complexity increases, and are a priori not tight.

The present manuscript aims at complementing and furthering this last body of works, by providing a tight end-to-end analysis of a flow-based generative model – starting from the study of the high-dimensional learning problem with a finite number of samples, and subsequently elucidating the implications thereof on the generative process.

Main contributions– We study the problem of estimating and sampling a Gaussian mixture using a flow-based generative model, in the framework of stochastic interpolation (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Lipman et al., 2022; Liu et al., 2022). We consider the case where a non-linear two-layer DAE with one hidden unit is used to parametrize the velocity field of the associated flow, and is trained with a finite training set. In the high-dimensional limit,

- We provide a sharp asymptotic closed-form characterization of the learnt velocity field, as a function of the target Gaussian mixture parameters, the stochastic interpolation schedule, and the number of training samples n.
- We characterize the associated flow by providing a tight characterization of a small number of summary statistics, tracking the dynamics of a sample from the Gaussian base distribution as it is transported by the learnt velocity field.
- We show that even with a finite number of training samples, the learnt generative model allows to sample from a mixture whose mean asymptotically approaches the mean of the target mixture as $\Theta_n(1/n)$ in squared distance, with this rate being tight.
- Finally, we show that this rate is in fact Bayes-optimal.

The code used in the present manuscript is provided in this repository.

Related works

Diffusion and flow-based generative models Score-based diffusion models (Song & Ermon, 2019; Song et al., 2020; Karras et al., 2022; Ho et al., 2020) build on the idea that any density can be mapped to a Gaussian density by degrading samples through an Ornstein-Uhlenbeck process. Sampling from the original density can then be carried out by time-reversing the corresponding stochastic transport, provided the score is known – or estimated. These ideas were subsequently refined in (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Lipman et al., 2022; Liu et al., 2022), which provide a flexible framework to bridge between two arbitrary densities in finite time.

Convergence bounds In the wake of the practical successes of flow and diffusion-based generative models, significant theoretical effort has been devoted to studying the convergence of such methods, by bounding appropriate distances between the generated and the target densities. A common assumption of (Benton et al., 2023; Chen et al., 2022; 2023a;c;d; Wibisono & Yang, 2022; Lee et al., 2022; 2023; Li et al., 2023; De Bortoli et al., 2021; De Bortoli, 2022; Pidstrigach, 2022; Block et al., 2020) is the availability of a good estimate for the score, i.e. an estimate whose average (population) squared distance with the true score is bounded by a small constant ϵ. Under this assumption, Chen et al. (2022); Lee et al. (2022) obtain rigorous control on the Wasserstein and total variation distances with very mild assumptions on the target density. Ghio et al. (2023) explore the connections between algorithmic hardness of the score/flow approximation and the hardness of sampling in a number of graphical models.

Asymptotics for DAE learning The backbone of flow and diffusion-based generative models is the parametrization of the score or velocity by a denoiser-type network, whose most standard realization is arguably the DAE (Vincent et al., 2010; Vincent, 2011). Very recent works have provided a detailed analysis of its learning on denoising tasks, for data sampled from Gaussian mixtures. Cui & Zdeborová (2023) sharply characterize how a DAE can learn the mixture parameters with $n = \Theta_d(d)$ training samples when the cluster separation is $\Theta_d(1)$. Closer to our work, for arbitrary cluster separation, Shah et al. (2023) rigorously show that a DAE trained with gradient descent on the denoising diffusion probabilistic model loss (Ho et al., 2020) can recover the cluster means with a polynomial number of samples. While these works complement the aforediscussed

convergence studies in that they analyze the effect of a finite number of samples, neither explores the flow associated to the learnt score.

Network-parametrized models Tying together these two body of works, a very recent line of research has addressed the problem of bounding, at finite sample complexity, appropriate distances between the generated and target densities, assuming a network-based parametrization. Oko et al. (2023) provide such bounds when parametrizing the score using a class of ReLU networks. These bounds however suffer from the curse of dimensionality. Oko et al. (2023); Yuan et al. (2023); Chen et al. (2023b) surmount this hurdle by assuming a target density with low-dimensional structure. On a heuristic level, Biroli & Mézard (2023) estimate the order of magnitude of the sample complexity needed to sample from a high-dimensional Curie-Weiss model. Finally, a work concurrent to ours (Mei & Wu, 2023) derives rigorous bounds for a number of high-dimensional graphical models. On the other hand, these bounds are a priori not tight, and do not go to zero as the sample complexity becomes large. The present manuscript aims at furthering this line of work, and provides a *sharp* analysis of a high-dimensional flow-based generative model.

1 SETTING

We start by giving a concise overview of the problem of sampling from a target density ρ_1 over $\mathbb{R}^d$ in the framework of stochastic interpolation (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023).

Recasting sampling as an optimization problem Samples from ρ_1 can be generated by drawing a sample from an easy-to-sample base density ρ_0 –henceforth taken to be a standard Gaussian density $\rho_0 = \mathcal{N}(0, \mathbb{I}_d)$–, and evolving it according to the flow described by the ordinary differential equation (ODE)

$$\frac{d}{dt}\boldsymbol{X}_t = \boldsymbol{b}(\boldsymbol{X}_t, t), \tag{1}$$

for $t \in [0, 1]$. Specifically, as shown in Albergo et al. (2023), if $\boldsymbol{X}_{t=0} \sim \rho_0$, then the final sample $\boldsymbol{X}_{t=1}$ has probability density ρ_1, if the velocity field $\boldsymbol{b}(\boldsymbol{x}, t)$ governing the flow (1) is given by

$$\boldsymbol{b}(\boldsymbol{x}, t) = \mathbb{E}[\dot{\alpha}(t)\boldsymbol{x}_0 + \dot{\beta}(t)\boldsymbol{x}_1 | \boldsymbol{x}_t = \boldsymbol{x}], \tag{2}$$

where we denoted $\boldsymbol{x}_t \equiv \alpha(t)\boldsymbol{x}_0 + \beta(t)\boldsymbol{x}_1$ and the conditional expectation bears over $\boldsymbol{x}_1 \sim \rho_1$, $\boldsymbol{x}_0 \sim \rho_0$, with $\boldsymbol{x}_0 \perp \boldsymbol{x}_1$. The result holds for any fixed choice of schedule functions $\alpha, \beta \in \mathcal{C}^2([0, 1])$ satisfying $\alpha(0) = \beta(1) = 1, \alpha(1) = \beta(0) = 0$, and $\alpha(t)^2 + \beta(t)^2 > 0$ for all $t \in [0, 1]$. In addition to the velocity field $\boldsymbol{b}(\boldsymbol{x}, t)$, it is convenient to consider the field $\boldsymbol{f}(\boldsymbol{x}, t)$, related to $\boldsymbol{b}(\boldsymbol{x}, t)$ by the simple relation

$$\boldsymbol{b}(\boldsymbol{x}, t) = \left(\dot{\beta}(t) - \frac{\dot{\alpha}(t)}{\alpha(t)}\beta(t)\right)\boldsymbol{f}(\boldsymbol{x}, t) + \frac{\dot{\alpha}(t)}{\alpha(t)}\boldsymbol{x}. \tag{3}$$

Note that $\boldsymbol{f}(\boldsymbol{x}, t)$ can be alternatively expressed as $\mathbb{E}[\boldsymbol{x}_1 | \boldsymbol{x}_t = \boldsymbol{x}]$, and thus admits a natural interpretation as a *denoising* function, tasked with recovering the target value $\boldsymbol{x}_1$ from the interpolated (noisy) sample $\boldsymbol{x}_t$. The denoiser $\boldsymbol{f}(\boldsymbol{x}, t)$ can furthermore characterized as the minimizer of the objective

$$\mathcal{R}[\boldsymbol{f}] = \int_0^1 \mathbb{E}\,\|\boldsymbol{f}(\boldsymbol{x}_t, t) - \boldsymbol{x}_1\|^2\, dt. \tag{4}$$

The loss (4) is a simple sequence of quadractic *denoising* objectives.

Learning the velocity from data There are several technical hurdles in carrying out the minimization (4). First, since the analytical form of ρ_1 is generically unknown, the population risk has to be approximated by its empirical version, provided a dataset $\mathcal{D} = \{\boldsymbol{x}_1^\mu, \boldsymbol{x}_0^\mu\}_{\mu=1}^n$ of n training samples $\boldsymbol{x}_1^\mu$ ($\boldsymbol{x}_0^\mu$) independently drawn from ρ_1 (ρ_0) is available. Second, the minimization in (4) bears over a time-dependent vector field $\boldsymbol{f}$. To make the optimization tractable, the latter can be parametrized at each time step t by a separate neural network $\boldsymbol{f}_{\theta_t}(\cdot)$ with trainable parameters θ_t. Under those approximations, the population risk (4) thus becomes

$$\hat{\mathcal{R}}(\{\theta_t\}_{t\in[0,1]}) = \int_0^1 \sum_{\mu=1}^n \|\boldsymbol{f}_{\theta_t}(\boldsymbol{x}_t^\mu) - \boldsymbol{x}_1^\mu\|^2\, dt. \tag{5}$$

Remark that in practice, the time t can enter as an input of the neural network, and only one network then needs to be trained. In the present manuscript however, for technical reasons, we instead consider the case where a *separate* network is trained *for each time step* t. Besides, note that since the base density ρ_0 is a priori easy to sample from, one could in theory augment the dataset $\mathcal{D}$ with several samples from ρ_0 for each available $\boldsymbol{x}_1^\mu$. For conciseness, we do not examine such an augmentation technique in the present manuscript, and leave a precise investigation thereof to future work. Denoting by $\{\hat{\theta}_t\}_{t\in[0,1]}$ the minimizer of (5), the learnt velocity field $\hat{\boldsymbol{b}}$ is related to the trained denoiser $\boldsymbol{f}_{\hat{\theta}_t}$ by (4) as

$$\hat{\boldsymbol{b}}(\boldsymbol{x},t) = \left(\dot{\beta}(t) - \frac{\dot{\alpha}(t)}{\alpha(t)}\beta(t)\right)\boldsymbol{f}_{\hat{\theta}_t}(\boldsymbol{x}) + \frac{\dot{\alpha}(t)}{\alpha(t)}\boldsymbol{x}. \tag{6}$$

The sampling can finally be carried out by using $\hat{\boldsymbol{b}}$ as a proxy for the unknown $\boldsymbol{b}$ in (1):

$$\frac{d}{dt}\boldsymbol{X}_t = \hat{\boldsymbol{b}}(\boldsymbol{X}_t, t) \tag{7}$$

Note that the solution $\boldsymbol{X}_1$ at time $t = 1$ of the ODE (7) has a law $\hat{\rho}_1 \neq \rho_1$ due to the two approximations in going from the population function-space objective (4) to the empirical parametric proxy (5). The present manuscript presents a sharp analysis of the learning problem (5) and the resulting flow (7) for a solvable model, which we detail below.

Data model We consider the case of a target density ρ_1 given by a binary isotropic and homoscedastic Gaussian mixture

$$\rho_1 = \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbb{I}_d) + \frac{1}{2}\mathcal{N}(-\boldsymbol{\mu}, \sigma^2\mathbb{I}_d). \tag{8}$$

Each cluster is thus centered around its mean $\pm\boldsymbol{\mu}$ and has variance σ^2. For definiteness, we consider here a balanced mixture, where the two clusters have equal relative probabilities, and defer the discussion of the imbalanced case to Appendix D. Note that a sample $\boldsymbol{x}_1^\mu$ can then be decomposed as $\boldsymbol{x}_1^\mu = s^\mu\boldsymbol{\mu} + \boldsymbol{z}^\mu$, with $s^\mu \sim \mathcal{U}(\{-1,+1\})$ and $\boldsymbol{z}^\mu \sim \mathcal{N}(0, \sigma^2\mathbb{I}_d)$. Finally, note that the closed-form expression for the exact velocity field $\boldsymbol{b}$ (1) associated to the density ρ_1 is actually known (see e.g. Efron (2011); Albergo et al. (2023)). This manuscript explores the question whether a neural network can learn a good approximate $\hat{\boldsymbol{b}}$ thereof *without* any knowledge of the density ρ_1, and only from a finite number of samples drawn therefrom.

Network architecture We consider the case where the denoising function $\boldsymbol{f}$ (4) is parametrized with a two-layer non-linear DAE with one hidden neuron, and –taking inspiration from modern practical architectures such as U-nets (Ronneberger et al., 2015)– a trainable skip connection:

$$\boldsymbol{f}_{\boldsymbol{w}_t,c_t}(\boldsymbol{x}) = c_t \times \boldsymbol{x} + \boldsymbol{w}_t \times \varphi(\boldsymbol{w}_t^\top\boldsymbol{x}), \tag{9}$$

where φ is assumed to tend to 1 (resp. -1) as its argument tends to $+\infty$ (resp $-\infty$). Sign, tanh and erf are simple examples of such an activation function. The trainable parameters are therefore $c_t \in \mathbb{R}, \boldsymbol{w}_t \in \mathbb{R}^d$. Note that (9) is a special case of the architecture studied in Cui & Zdeborová (2023). It differs from the very similar network considered in Shah et al. (2023) in that it covers a slightly broader range of activation functions (Shah et al. (2023) address the case $\varphi = \tanh$), and in that the skip connection istrainable –rather than fixed–. Since we consider the case where a separate network is trained at every time step, the empirical risk (5) decouples over the time index t. The parameters $\boldsymbol{w}_t, c_t$ of the DAE (9) should therefore minimize

$$\hat{\mathcal{R}}_t(\boldsymbol{w}_t, c_t) = \sum_{\mu=1}^{n}\|\boldsymbol{f}_{c_t,\boldsymbol{w}_t}(\boldsymbol{x}_t^\mu) - \boldsymbol{x}_1^\mu\|^2 + \frac{\lambda}{2}\|\boldsymbol{w}_t\|^2, \tag{10}$$

where for generality we also allowed for the presence of a ℓ_2 regularization of strength λ. We remind that $\boldsymbol{x}_t^\mu = \alpha(t)\boldsymbol{x}_0^\mu + \beta(t)\boldsymbol{x}_1^\mu$, with $\{\boldsymbol{x}_1^\mu\}_{\mu=1}^n$ (resp. $\{\boldsymbol{x}_0^\mu\}_{\mu=1}^n$) n training samples independently drawn from the target density ρ_1 (8) (resp. the base density $\rho_0 = \mathcal{N}(0, \mathbb{I}_d)$), collected in the training set $\mathcal{D}$.

Asymptotic limit We consider in this manuscript the asymptotic limit $d \to \infty$, with $n, \|\boldsymbol{\mu}\|^2/d, \sigma = \Theta_d(1)$. For definiteness, in the following, we set $\|\boldsymbol{\mu}\|^2/d = 1$. Note that Cui & Zdeborová (2023) consider the different limit $\|\boldsymbol{\mu}\| = \Theta_d(1)$. Shah et al. (2023) on the other hand address a larger range of asymptotic limits, including the present one, but does not provide tight characterizations, nor an analysis of the generative process.

2 LEARNING

In this section, we first provide sharp closed-form characterizations of the minimizers $\hat{c}_t, \hat{\boldsymbol{w}}_t$ of the objective $\hat{\mathcal{R}}_t$ (10). The next section discusses how these formulae can be leveraged to access a tight characterization of the associated flow.

Result 2.1. ***(Sharp characterization of minimizers of (10))*** *For any given activation φ satisfying $\varphi(x) \xrightarrow{x\to\pm\infty} \pm 1$ and any $t \in [0,1]$, in the limit $d \to \infty$, $n, \|\boldsymbol{\mu}\|^2/d, \sigma = \Theta_d(1)$, the skip connection strength $\hat{c}_t$ minimizing (10) is given by*

$$\hat{c}_t = \frac{\beta(t)(\lambda(1+\sigma^2)+(n-1)\sigma^2)}{\alpha(t)^2(\lambda+n-1)+\beta(t)^2(\lambda(1+\sigma^2)+(n-1)\sigma^2)}. \tag{11}$$

Furthermore, the learnt weight vector $\hat{\boldsymbol{w}}_t$ is asymptotically contained in $\mathrm{span}(\boldsymbol{\mu}_{\mathrm{emp.}}, \boldsymbol{\xi})$ (in the sense that its projection on the orthogonal space $\mathrm{span}(\boldsymbol{\mu}_{\mathrm{emp.}}, \boldsymbol{\xi})$ has asymptotically vanishing norm), where

$$\boldsymbol{\xi} \equiv \sum_{\mu=1}^{n} s^\mu \boldsymbol{x}_0^\mu, \qquad\qquad \boldsymbol{\mu}_{\mathrm{emp.}} = \frac{1}{n}\sum_{\mu=1}^{n} s^\mu x_1^\mu. \tag{12}$$

In other words, $\boldsymbol{\mu}_{\mathrm{emp.}}$ is the empirical mean of the training samples. We remind that $s^\mu = \pm 1$ was defined below (8) and indicates the cluster the $\mu-$th sample $\boldsymbol{x}_1^\mu$ belongs to. The components of $\hat{\boldsymbol{w}}_t$ along each of these three vectors is described by the summary statistics

$$m_t = \frac{\boldsymbol{\mu}_{\mathrm{emp.}}^\top \hat{\boldsymbol{w}}_t}{d(1+\sigma^2/n)}, \qquad\qquad q_t^\xi = \frac{\hat{\boldsymbol{w}}_t^\top \boldsymbol{\xi}}{nd}, \tag{13}$$

which concentrate as $d \to \infty$ to the quantities characterized by the closed-form formulae

$$\begin{cases} m_t = \frac{n}{\lambda+n}\frac{\alpha(t)^2(\lambda+n-1)}{\alpha(t)^2(\lambda+n-1)+\beta(t)^2(\lambda(1+\sigma^2)+(n-1)\sigma^2)} \\ q_t^\xi = \frac{-\alpha(t)}{\lambda+n}\frac{\beta(t)(\lambda(1+\sigma^2)+(n-1)\sigma^2)}{\alpha(t)^2(\lambda+n-1)+\beta(t)^2(\lambda(1+\sigma^2)+(n-1)\sigma^2)} \end{cases}. \tag{14}$$

The derivation of Result 2.1 is detailed in Appendix A, and involves a heuristic partition function computation, borrowing ideas from statistical physics. The theoretical predictions for the skip connection strength $\hat{c}_t$ and the component m_t, q_t^ξ of the weight vector $\hat{\boldsymbol{w}}_t$ are plotted as solid lines in Fig. 1, and display good agreement with numerical simulations, corresponding to training the DAE (9) on the risk (10) using the `Pytorch` (Paszke et al., 2019) implementation of the Adam optimizer (Kingma & Ba, 2014).

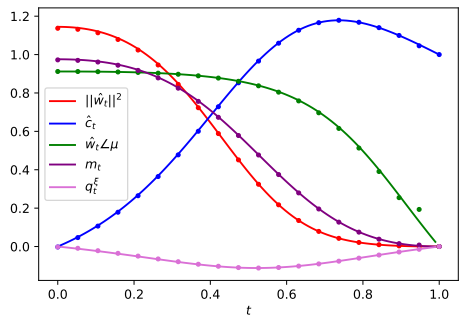

Figure 1: $n = 4, \sigma = 0.9, \lambda = 0.1, \alpha(t) = 1-t, \beta(t) = t, \varphi = \tanh$. Solid lines: theoretical predictions of Result 2.1: squared norm of the DAE weight vector $\|\hat{\boldsymbol{w}}_t\|^2$ (red), skip connection strength $\hat{c}_t$ (blue) cosine similarity between the weight vector $\hat{\boldsymbol{w}}_t$ and the target cluster mean $\boldsymbol{\mu}$, $\hat{\boldsymbol{w}}_t \angle \boldsymbol{\mu} \equiv \hat{\boldsymbol{w}}_t^\top \boldsymbol{\mu}/\|\boldsymbol{\mu}\|\|\hat{\boldsymbol{w}}_t\|$ (green), components m_t, q_t^ξ of $\hat{\boldsymbol{w}}_t$ along the vectors $\boldsymbol{\mu}_{\mathrm{emp.}}, \boldsymbol{\xi}$ (purple, pink, orange). Dots: numerical simulations in dimension $d = 5 \times 10^4$, corresponding to training the DAE (9) on the risk (10) using the `Pytorch` implementation of full-batch Adam, with learning rate 0.0001 over 4×10^4 epochs and weight decay $\lambda = 0.1$. The experimental points correspond to a single instance of the model.

A notable consequence of (13) is that the weight vector $\hat{\boldsymbol{w}}_t$ is contained at all times t in the two-dimensional subspace spanned by the empirical cluster mean $\boldsymbol{\mu}_{\mathrm{emp.}}$ and the vectors $\boldsymbol{\xi}$ (12) – in other words, the learnt weights align to some extent with the empirical mean, but still possess a non-zero component along $\boldsymbol{\xi}$, which is orthogonal thereto. $\boldsymbol{\xi}$ subsumes the aggregated effect of the base vectors $\{\boldsymbol{x}_0^\mu\}_{\mu=1}^n$ used in the train set. Rather remarkably, the training samples thus only enter in the characterization of $\hat{\boldsymbol{w}}_t$ through the form of simple sums (12). Since the vector $\boldsymbol{\xi}$ is associated to the training samples, the fact that the learnt vector $\hat{\boldsymbol{w}}_t$ has non-zero components along $\boldsymbol{\xi}$ hence signals a form of overfitting and memorization. Interestingly, Fig. 1 shows that the extent of this overfitting is non-monotonic

in time, as $|q_t^\xi|$ first increases then decreases. Finally, note that this effect is as expected mitigated as the number of training samples n increases. From (14), for large n, $m_t = \Theta_n(1)$ while the components q_t^ξ is suppressed as $\Theta_n(1/n)$. These scalings are further elaborated upon in Remark B.3 in Appendix B. Finally, Result 2.1 and equation (6) can be straightforwardly combined to yield a sharp characterization of the learnt estimate $\hat{\boldsymbol{b}}$ of the velocity field $\boldsymbol{b}$ (1). This characterization can be in turn leveraged to build a tight description of the generative flow (7). This is the object of the following section.

3 GENERATIVE PROCESS

While Corollary 2.1, together with the definition (6), provides a concise characterization of the velocity field $\hat{\boldsymbol{b}}$, the sampling problem (7) remains formulated as a high-dimensional, and therefore hard to analyze, transport process. The following result shows that the dynamics of a sample $\boldsymbol{X}_t$ following the differential equation (7) can nevertheless be succinctly tracked using a finite number of scalar summary statistics.

Result 3.1. ***(Summary statistics)*** *Let $\boldsymbol{X}_t$ be a solution of the ordinary differential equation (7) with initial condition $\boldsymbol{X}_0$. For a given t, the projection of $\boldsymbol{X}_t$ on $\mathrm{span}(\boldsymbol{\mu}_{\mathrm{emp.}}, \boldsymbol{\xi}$ is characterized by the summary statistics*

$$M_t \equiv \frac{\boldsymbol{X}_t^\top \boldsymbol{\mu}_{\mathrm{emp.}}}{d(1+\sigma^2/n)}, \qquad\qquad Q_t^\xi \equiv \frac{\boldsymbol{X}_t^\top \boldsymbol{\xi}}{nd}. \tag{15}$$

With probability asymptotically $1/2$ the summary statistics M_t, Q_t^ξ (15) concentrate for all t to the solution of the ordinary differential equations

$$\begin{cases} \frac{d}{dt} M_t = \left(\dot\beta(t)\hat c_t + \frac{\dot\alpha(t)}{\alpha(t)}(1-\hat c_t \beta(t))\right) M_t + \left(\dot\beta(t) - \frac{\dot\alpha(t)}{\alpha(t)}\beta(t)\right) m_t \\ \frac{d}{dt} Q_t^\xi = \left(\dot\beta(t)\hat c_t + \frac{\dot\alpha(t)}{\alpha(t)}(1-\hat c_t \beta(t))\right) Q_t^\xi + \left(\dot\beta(t) - \frac{\dot\alpha(t)}{\alpha(t)}\beta(t)\right) q_t^\xi \end{cases}, \tag{16}$$

with initial condition $M_0 = Q_0^\xi = 0$, and with probability asymptotically $1/2$ they concentrate to minus the solution of (16). Furthermore, the orthogonal component $\boldsymbol{X}_t^\perp \in \mathrm{span}(\boldsymbol{\mu}_{\mathrm{emp.}}, \boldsymbol{\xi})^\perp$ obeys the simple linear differential equation

$$\frac{d}{dt}\boldsymbol{X}_t^\perp = \left(\dot\beta(t)\hat c_t + \frac{\dot\alpha(t)}{\alpha(t)}(1-\hat c_t\beta(t))\right)\boldsymbol{X}_t^\perp. \tag{17}$$

Finally, the statistic $Q_t \equiv \|\boldsymbol{X}_t\|^2/d$ is given with high probability by

$$Q_t = M_t^2(1+\sigma^2/n) + n(Q_t^\xi)^2 + e^{2\int_0^t \left(\dot\beta(t)\hat c_t + \frac{\dot\alpha(t)}{\alpha(t)}(1-\hat c_t\beta(t))\right)dt}. \tag{18}$$

A heuristic derivation of Result 3.1 is provided in Appendix B. Taking a closer look at (16), it might seem at first from equations (16) that there is a singularity for $t = 1$ since $\alpha(1) = 0$ in the denominator. Remark however that both $1 - \beta(t)\hat c_t$ (11) and m_t (14) are actually proportional to $\alpha(t)^2$, and therefore (16) is in fact also well defined for $t = 1$. In practice, the numerical implementation of a generative flow like (7) often involves a discretization thereof, given a discretization scheme $\{t_k\}_{k=0}^N$ of $[0,1]$, where $t_0 = 0$ and $t_N = 1$:

$$\boldsymbol{X}_{t_{k+1}} = \boldsymbol{X}_{t_k} + \hat{\boldsymbol{b}}(\boldsymbol{X}_{t_k}, t_k)(t_{k+1} - t_k). \tag{19}$$

The evolution of the summary statistics introduced in Result 3.1 can be rephrased in more actionable form to track the discretized flow (19).

Remark 3.2. ***(Summary statistics for the discrete flow)*** *Let $\{\boldsymbol{X}_{t_k}\}_{k=0}^N$ be a solution of the discretized learnt flow (7), for an arbitrary discretization scheme $\{t_k\}_{k=0}^N$ of $[0,1]$, where $t_0 = 0$ and $t_N = 1$, with initial condition $\boldsymbol{X}_{t_0} \sim \rho_0$. The summary statistics introduced in Result 3.1 are then equal to the solutions of the recursions*

$$\begin{cases} M_{t_{k+1}} = M_{t_k} + \delta t_k \left(\dot\beta(t_k)\hat c_{t_k} + \frac{\dot\alpha(t_k)}{\alpha(t_k)}(1-\hat c_{t_k}\beta(t_k))\right) M_{t_k} + \delta t_k \left(\dot\beta(t_k) - \frac{\dot\alpha(t_k)}{\alpha(t_k)}\beta(t_k)\right) m_{t_k} \\ Q_{t_{k+1}}^\xi = Q_{t_k}^\xi + \delta t_k \left(\dot\beta(t_k)\hat c_{t_k} + \frac{\dot\alpha(t_k)}{\alpha(t_k)}(1-\hat c_{t_k}\beta(t_k))\right) Q_{t_k}^\xi + \delta t_k \left(\dot\beta(t_k) - \frac{\dot\alpha(t_k)}{\alpha(t_k)}\beta(t_k)\right) q_{t_k}^\xi \end{cases}, \tag{20}$$

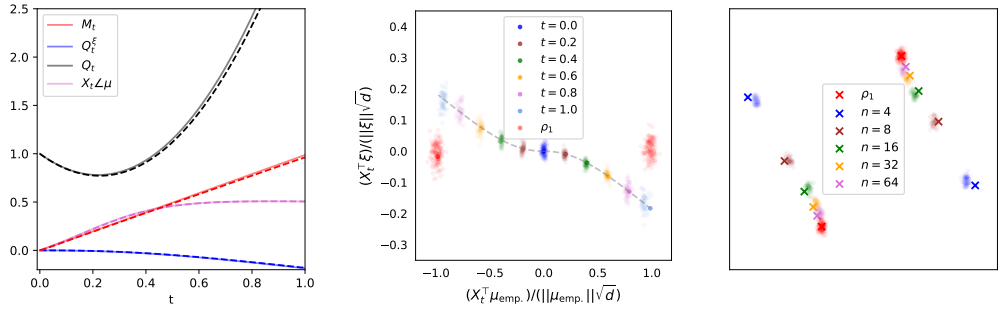

Figure 2: In all three plots, $\lambda = 0.1, \alpha(t) = 1 - t, \beta(t) = t, \varphi = \text{sign}$. **(left)** $\sigma = 1.5, n = 8$. Temporal evolution of the summary statistics $M_t, Q_t^\xi, Q_t, \boldsymbol{X}_t \angle \boldsymbol{\mu}$ (15). Solid lines correspond to the theoretical prediction of (15) in Result 3.1, while dashed lines correspond to numerical simulations of the generative model, by discretizing the differential equation (7) with step size $\delta t = 0.01$, and training a separate DAE for each time step using Adam with learning rate 0.01 for 2000 epochs. All experiments were conducted in dimension $d = 5000$, and a single run is represented. **(middle)** $\sigma = 2, n = 16$. Projection of the distribution of $\boldsymbol{X}_t$ (7) in $\text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})$, transported by the velocity field $\hat{\boldsymbol{b}}$ (6) learnt from data. The point clouds correspond to numerical simulations. The dashed line corresponds to the theoretical prediction of the means of the cluster, as given by equation (16) of Result 3.1. The target Gaussian mixture ρ_1 is represented in red. The base zero-mean Gaussian density ρ_0 (dark blue) is split by the flow (7) into two clusters, which approach the target clusters (red) as time accrues . **(right)** $\sigma = 2$. PCA visualization of the generated density $\hat{\rho}_1$, by training the generative model on n samples, for $n \in \{4, 8, 16, 32, 64\}$. Point clouds represent numerical simulations of the generative model. Crosses represent the theoretical predictions of Result 3.1 for the means of the clusters of $\hat{\rho}_1$, as given by equation (16) of Result 3.1 for $t = 1$. As the number of training samples n increases, the generated clusters of $\hat{\rho}_1$ approach the target clusters of ρ_1, represented in red.

with probability $1/2$, and to the opposite theoreof with probability $1/2$. In equation (20), the initial conditions are understood as $M_{t_0} = Q_{t_0}^\xi = 0$, and we have denoted $\delta t_k \equiv t_{k+1} - t_k$ for clarity. Furthermore, the orthogonal component $\boldsymbol{X}_{t_k}^\perp \in \text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})^\perp$ obeys the simple linear recursion

$$\boldsymbol{X}_{t_{k+1}}^\perp = \left[1 + \delta t_k \left(\dot{\beta}(t_k)\hat{c}_{t_k} + \frac{\dot{\alpha}(t_k)}{\alpha(t_k)}(1 - \hat{c}_{t_k}\beta(t_k))\right)\right] \boldsymbol{X}_{t_k}^\perp. \tag{21}$$

Finally, the statistic $Q_{t_k} \equiv \|\boldsymbol{X}_{t_k}\|^2/d$ is given with high probability by

$$Q_{t_k} = M_{t_k}^2(1 + \sigma^2/n) + n(Q_{t_k}^\xi)^2 + \prod_{\ell=0}^{k} \left[1 + \left(\dot{\beta}(t_\ell)\hat{c}_{t_\ell} + \tfrac{\dot{\alpha}(t_\ell)}{\alpha(t_\ell)}(1 - \hat{c}_{t_\ell}\beta(t_\ell))\right)\delta t_\ell\right]^2. \tag{22}$$

Equations (20),(21) and (22) of Remark 3.2 are consistent discretizations of the continuous flows (16),(17) and (18) of Result 3.1 respectively, and converge thereto in the limit of small discretization steps $\max_k \delta t_k \to 0$. A derivation of Remark 3.2 is detailed in Appendix B. An important consequence of Result 3.1 is that the transport of a sample $\boldsymbol{X}_0 \sim \rho_0$ by (7) factorizes into the low-dimensional deterministic evolution of its projection on the low-rank subspace $\text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})$, as tracked by the two summary statistics M_t, Q_t^ξ, and the simple linear dynamics of its projection on the orthogonal space $\text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})^\perp$. Result 3.1 thus reduces the high-dimensional flow (7) into a set of two scalar ordinary differential equations (16) and a simple homogeneous linear differential equation (17). The theoretical predictions of Result (3.1) and Remark 3.2 for the summary statistics M_t, Q_t^ξ, Q_t are plotted in Fig. 2, and display convincing agreement with numerical simulations, corresponding to discretizing the flow (7) in $N = 100$ time steps, and training a separate network for each step as described in Section 1. A PCA visualization of the flow is further provided in Fig. 2 (middle).

Leveraging the simple characterization of Result 3.1, one is now in a position to characterize the generated distribution $\hat{\rho}_1$, which is the density effectively sampled by the generative model. In particular, Result 3.1 establishes that the distribution $\hat{\rho}_1$ is Gaussian over $\text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})^\perp$ – since $\boldsymbol{X}_0^\perp$ is Gaussian and the flow is linear–, while the density in $\text{span}(\boldsymbol{\mu}_{\text{emp.}}, \boldsymbol{\xi})$ concentrates along the vector $\hat{\boldsymbol{\mu}}$ described by the components (16). The density $\hat{\rho}_1$ is thus described by a mixture of two

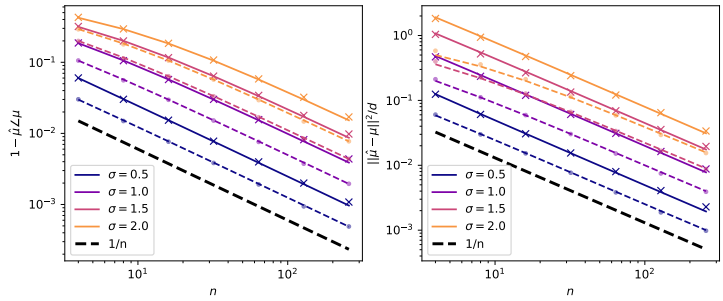

Figure 3: $\alpha(t) = 1 - t, \beta(t) = t, \varphi =$ sign. Cosine asimilarity (left) and mean squared distance (right) between the mean $\hat{\boldsymbol{\mu}}$ of the generated mixture $\hat{\rho}_1$ and the mean $\boldsymbol{\mu}$ of the target density ρ_1, as a function of the number of training samples n, for various variances σ of ρ_1. Solid lines represent the theoretical characterization of Corollary 3.3. Crosses represent numerical simulations of the generative model, by discretizing the differential equation (7) with step size $\delta t = 0.01$, and training a separate DAE for each time step using the `Pytorch` implementation of the full-batch Adam optimizer, with learning rate 0.04 and weight decay $\lambda = 0.1$ for 6000 epochs. All experiments were conducted in dimension $d = 5 \times 10^4$, and a single run is represented. Dashed lines indicate the performance of the Bayes-optimal estimator $\hat{\boldsymbol{\mu}}^\star$, as theoretically characterized in Remark 4.1. Dots indicate the performance of the PCA estimator, which is found as in Cui & Zdeborová (2023) to yield performances nearly identical to the Bayes-optimal estimator.

clusters, Gaussian along $d-2$ directions, centered around $\pm\hat{\boldsymbol{\mu}}$. The following corollary provides a sharp characterization of the squared distance between the mean $\hat{\boldsymbol{\mu}}$ of the generated density $\hat{\rho}_1$ and the true mean $\boldsymbol{\mu}$ of the target density ρ_1.

Corollary 3.3. ***(Mean squared error of the mean estimate)*** *Let $\hat{\boldsymbol{\mu}}$ be the cluster mean of the density $\hat{\rho}_1$ generated by the (continuous) learnt flow (7). In the asymptotic limit described by Result 2.1, the squared distance between $\hat{\boldsymbol{\mu}}$ and the true mean $\boldsymbol{\mu}$ is given by*

$$\frac{1}{d}\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 = M_1^2 + n(Q_1^\xi)^2 + n\sigma^2(Q_1^\eta)^2 + 1 - 2M_1, \tag{23}$$

with M_1, Q_1^ξ, Q_1^η being the solutions of the ordinary differential equations (16) evaluated at time $t = 1$. Furthermore, the cosine similarity between $\hat{\boldsymbol{\mu}}$ and the true mean $\boldsymbol{\mu}$ is given by

$$\hat{\boldsymbol{\mu}}\angle\boldsymbol{\mu} = \frac{M_1}{\sqrt{Q_1}}. \tag{24}$$

Finally, both the Mean Squared Error (MSE) $1/d\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2$ (23) and the cosine asimilarity $1 - \hat{\boldsymbol{\mu}}\angle\boldsymbol{\mu}$ (24) decay as $\Theta_n(1/n)$ for large number of samples n.

The heuristic derivation of Corollary 3.3 is presented in Appendix A.1. The theoretical predictions of the learning metrics (23) and (24) are plotted in Fig. 3 as a function of the number of samples, along with the corresponding numerical simulations, and display a clear $\Theta_n(1/n)$ decay, signalling the convergence of the generated density $\hat{\rho}_1$ to the true target density ρ_1 as the sample complexity accrues. A PCA visualization of this convergence is further presented in Fig.2 (right). Intuitively, this is because the DAE learns the empirical means up to a $\Theta_n(1/n)$ component along $\boldsymbol{\xi}$, and that the empirical means itself converges to the true mean with rate $\Theta_n(1/n)$. While we focus on the MSE for conciseness, the rate of convergence in terms of a variant of the squared gaussian mixture Wasserstein distance (Delon & Desolneux, 2020; Chen et al., 2018) can similarly be derived to be $\Theta_n(1/n)$, see Appendix F.

4 Bayes-optimal baseline

Corollary 3.3 completes the study of the performance of the DAE-parametrized generative model. It is natural to wonder whether one can improve on the $\Theta_n(1/n)$ rate that it achieves. A useful baseline to compare with is the Bayes-optimal estimator $\hat{\boldsymbol{\mu}}^\star$, yielded by Bayesian inference when in addition to the dataset $\mathcal{D} = \{\boldsymbol{x}_1^\mu\}_{\mu=1}^n$, the form of the distribution (8) and the variance σ are known, but *not* the mean $\boldsymbol{\mu}$ –which for definiteness and without loss of generality will be assumed in this section to be have been drawn at random from $\mathcal{N}(0, \mathbb{I}_d)$. The following remark provides a tight characterization of the MSE achieved by this estimator.

Remark 4.1. ***(Bayes-optimal estimator of the cluster mean)*** *The Bayes-optimal estimator $\hat{\boldsymbol{\mu}}^\star$ of $\boldsymbol{\mu}$ assuming knowledge of the functional form of the target density (8), the cluster variance σ, and the training set $\mathcal{D}$, is defined as the minimizer of the average squared error*

$$\hat{\boldsymbol{\mu}}^\star = \underset{\nu}{\operatorname{arginf}} \mathbb{E}_{\boldsymbol{\mu}\sim\mathcal{N}(0,\mathbb{I}_d),\mathcal{D}\sim\rho_1^{\otimes n}} \|\nu(\mathcal{D}) - \boldsymbol{\mu}\|^2. \tag{25}$$

In the asymptotic limit of Result 2.1, the Bayes-optimal estimator $\hat{\boldsymbol{\mu}}^\star(\mathcal{D})$ is parallel to the empirical mean $\boldsymbol{\mu}_{\text{emp.}}$. Its component $m^\star \equiv \boldsymbol{\mu}_{\text{emp.}}^\top \hat{\boldsymbol{\mu}}^\star(\mathcal{D})/d(1+\sigma^2/n)$ concentrate asymptotically to

$$m^\star = \frac{n}{n+\sigma^2}, \tag{26}$$

Finally, with high probability, the Bayes-optimal MSE reads

$$\frac{1}{d}\|\hat{\boldsymbol{\mu}}^\star(\mathcal{D}) - \boldsymbol{\mu}\|^2 = \frac{\sigma^2}{n+\sigma^2}. \tag{27}$$

In particular, (27) implies that the optimal MSE decays as $\Theta_n(1/n)$.

Remark 4.1, whose derivation is detailed in Appendix C, thus establishes that the Bayes-optimal MSE decays as $\Theta_n(1/n)$ with the number of available training samples. Note that while the Bayes-optimal estimator is colinear to the empirical mean, it is differs therefrom by a non-trivial multiplicative factor. On the other hand, the $\Theta_n(1/n)$ rate is intuitively due to the $\Theta_n(1/n)$ convergence of the empirical mean to the true mean. Contrasting to Corollary 3.3 for the MSE associated to the mean $\hat{\boldsymbol{\mu}}$ of the density $\hat{\rho}_1$ learnt by the generative model, it follows that *the latter achieves the Bayes-optimal learning rate*. The Bayes-optimal MSE (27) predicted by Remark 4.1 is plotted in dashed lines in Fig. 3, alongside the MSE achieved by the generative model (see Corollary 3.3). The common $1/n$ decay rate is also plotted in dashed black for comparison. Finally, we observe that the estimate of $\boldsymbol{\mu}$ inferred by PCA, plotted as dots in Fig. 3, leads to a cosine similarity which is very close to the Bayes-optimal one, echoing the findings of Cui & Zdeborová (2023) in another asymptotic limit. We however stress an important distinction between the generative model analyzed in previous sections and the Bayes and PCA estimators dicussed in the present section. The generative model is tasked with estimating the full distribution ρ_1 only from data, while being completely agnostic thereof. In contrast, PCA and Bayesian inference only offer an estimate of the cluster mean, and require an exact oracle knowledge of its functional form (8) and the cluster variance σ. They do *not*, therefore, constitute generative models and are only discussed in the present section as insightful baselines.

It is a rather striking finding that the DAE (9) succeeds in approximately sampling from $\rho_1(8)$ when trained on but $n = \Theta_d(1)$ samples –instead of simply generating back memorized training samples–, and further displays information-theoretically optimal learning rates. The answer to this puzzle lies in the fact that the architecture (9) is very close to the functional form of the exact velocity field b (1), as further detailed in Appendix B (see equation (67)), and is therefore implicitly biased towards learning the latter – while also not being expressive enough to too detrimentally overfit. A thorough exploration of this form of inductive bias for more complex architectures is an important and fascinating entreprise, which falls out of the scope of the present manuscript and is left for future work.

CONCLUSION

We conduct a tight end-to-end asymptotic analysis of estimating and sampling a binary Gaussian mixture using a flow-based generative model, when the flow is parametrized by a shallow auto-encoder. We provide sharp closed-form characterizations for the trained weights of the network, the learnt velocity field, a number of summary statistics tracking the generative flow, and the distance between the mean of the generated mixture and the mean of the target mixture. The latter is found to display a $\Theta_n(1/n)$ decay rate, where n is the number of samples, which is further shown to be the Bayes-optimal rate. In contrast to most studies of flow-based generative models in high dimensions, the learning and sampling processes are jointly and sharply analyzed in the present manuscript, which affords the possibility to explicitly investigate the effect of a limited sample complexity at the level of the generated density.

ACKNOWLEDGEMENT

We thank Michael Albergo, Nicholas Boffi, Joan Bruna, Arthur Jacot and Ahmed El Alaoui for insightful discussions. Part of this work was done during HC's visit in the Courant Institute in March

2023. We acknowledge funding from the Swiss National Science Foundation grants OperaGOST (grant number 200390) and SMArtNet (grant number 212049). EVE is supported by the National Science Foundation under awards DMR-1420073, DMS-2012510, and DMS-2134216, by the Simons Collaboration on Wave Turbulence, Grant No. 617006, and by a Vannevar Bush Faculty Fellowship.

REFERENCES

M. S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *ArXiv*, abs/2303.08797, 2023.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *arXiv preprint arXiv:2305.16860*, 2023.

Giulio Biroli and Marc Mézard. Generative diffusion in very large dimensions. *arXiv preprint arXiv:2306.03518*, 2023.

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

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*, pp. 4735–4763. PMLR, 2023a.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. *arXiv preprint arXiv:2302.07194*, 2023b.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *arXiv preprint arXiv:2305.11798*, 2023c.

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*, pp. 4462–4484. PMLR, 2023d.

Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

Hugo Cui and Lenka Zdeborová. High-dimensional asymptotics of denoising autoencoders. *arXiv preprint arXiv:2305.11041*, 2023.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106:1602 – 1614, 2011.

Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborová. Sampling with flows, diffusion and autoregressive neural networks: A spin-glass perspective. *arXiv preprint arXiv:2308.14085*, 2023.

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Yukito Iba. The nishimori line and bayesian statistics. *Journal of Physics A: Mathematical and General*, 32(21):3875, 1999.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pp. 946–985. PMLR, 2023.

Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. *arXiv preprint arXiv:2306.09251*, 2023.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Song Mei and Yuchen Wu. Deep networks as denoising algorithms: Sample-efficient learning of diffusion models in high-dimensional graphical models. *arXiv preprint arXiv:2309.11420*, 2023.

Hidetoshi Nishimori. *Statistical physics of spin glasses and information processing: an introduction*. Number 111. Clarendon Press, 2001.

Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. *arXiv preprint arXiv:2303.01861*, 2023.

Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Jakiw Pidstrigach. Score-based generative models detect manifolds. *Advances in Neural Information Processing Systems*, 35:35852–35865, 2022.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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*, pp. 234–241. Springer, 2015.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the ddpm objective. *arXiv preprint arXiv:2307.01178*, 2023.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Andre Wibisono and Kaylee Yingxi Yang. Convergence in kl divergence of the inexact langevin algorithm with application to score-based generative models. *arXiv preprint arXiv:2211.01512*, 2022.

Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. *arXiv preprint arXiv:2307.07055*, 2023.