Diffusion models for Gaussian distributions: Exact solutions and Wasserstein errors

Emile Pierret, Bruno Galerne

Diffusion models for Gaussian distributions: Exact solutions and Wasserstein errors

Emile Pierret
Institut Denis Poisson
Université d'Orléans
Orléans, France
emile.pierret@univ-orleans.fr

Bruno Galerne
Institut Denis Poisson
Université d'Orléans
Orléans, France
Institut universitaire de France (IUF)
bruno.galerne@univ-orleans.fr

Abstract

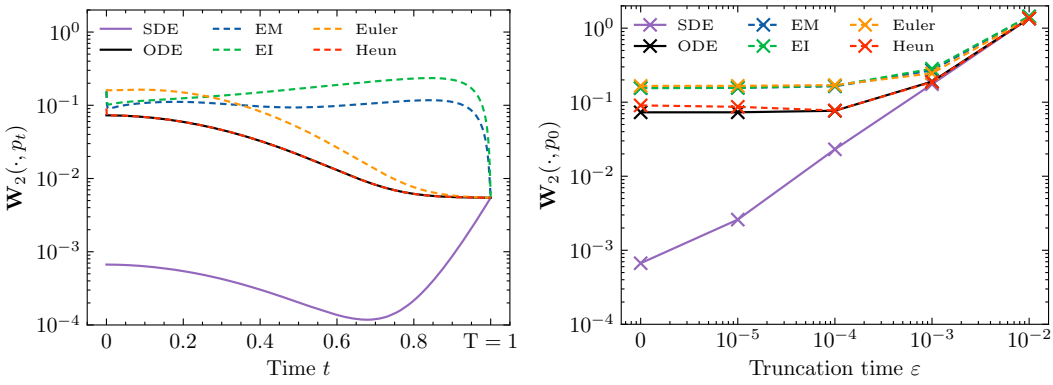
Diffusion or score-based models recently showed high performance in image generation. They rely on a forward and a backward stochastic differential equations (SDE). The sampling of a data distribution is achieved by solving numerically the backward SDE or its associated flow ODE. Studying the convergence of these models necessitates to control four different types of error: the initialization error, the truncation error, the discretization and the score approximation. In this paper, we study theoretically the behavior of diffusion models and their numerical implementation when the data distribution is Gaussian. In this restricted framework where the score function is a linear operator, we can derive the analytical solutions of the forward and backward SDEs as well as the associated flow ODE. This provides exact expressions for various Wasserstein errors which enable us to compare the influence of each error type for any sampling scheme, thus allowing to monitor convergence directly in the data space instead of relying on Inception features. Our experiments show that the recommended numerical schemes from the diffusion models literature are also the best sampling schemes for Gaussian distributions.

1 Introduction

Over the last five years, diffusion models have proven to be a highly efficient and reliable framework for generative modeling [25, 14, 24, 26, 7, 15]. First introduced as a discrete process, Denoising Diffusion Probabilistic Models (DDPM) [14] can be studied as a reversal of a continuous Stochastic Differential Equation (SDE) [26]. A forward SDE progressively transforms the initial data distribution by adding more and more noise as time grows. Then, the reversal of this process, called backward SDE, allows us to approximately sample the data distribution starting from Gaussian white noise. Moreover, the SDE is associated with an Ordinary Differential Equations (ODE) called probability flow [26]. This flow preserves the same marginal distributions as the backward SDE and provides another way to sample the score-based generative model.

An important issue about diffusion models is the theoretical guarantees of convergence of the model: How close to the data distribution the generate distribution is? There are four main source of errors to study for deriving theoretical guarantees for diffusion models. First, the *initialization error* is induced when approximating the marginal distribution at the end of the forward process by a standard Gaussian distribution. Then, the *discretization error* comes from the resolution of the SDE or the ODE by a numerical method. The *truncation error* occurs because the backward time integration is stopped at a small time $\varepsilon > 0$ to avoid numerical instabilities due to ill-defined score function near the origin. Finally the *score approximation error* accounts for the mismatch between the ideal score function and the one given by the network trained using denoising score-matching.

Preprint. Under review.



(a) Initialization error along the integration time

(b) Truncation error for different truncation time ε

Figure 1: **Wasserstein errors for the diffusion models associated with the CIFAR-10 Gaussian.** Left: Evolution of the Wasserstein distance between p_t and the distributions associated with the continuous SDE, the continuous flow ODE and four discrete sampling schemes with standard $\mathcal{N}_0$ initialization, either stochastic (Euler-Maruyama (EM) and Exponential Integrator (EI)) or deterministic (Euler and Heun). While the continuous SDE is less sensible than the continuous ODE (as proved by Proposition 4), the initialization error impacts all discrete schemes with a comparable order of magnitude. Heun's method has the lowest error and is very close to the theoretical ODE, except for the last step that is usually discarded when using time truncation. Right: Wasserstein errors due to time truncation for various truncation times ε. Using time truncation increases the error for all the methods except Heun's scheme due to instability near the origin. Interestingly, for the standard practice truncation time $\varepsilon = 10^{-3}$, all numerical schemes have a comparable error close to their continuous counterparts.

Despite these numerous sources of errors, a lot of numerical and theoretical research has been led to assess the generative capacity of diffusion models. Several articles [3, 10, 15] provide strong experimental studies for the choices of sampling parameters. On the theoretical side, several works derive upper bounds on the 1-Wasserstein or TV distance between the data and the model distributions by making assumptions on the L^2-error between the ideal and learned score functions and on the compacity of the support of the data [19, 6, 4, 18, 1], eventually under an additional manifold assumption [5, 28, 2]. Yet, on one hand, to the best of our knowledge, the derived theoretical bounds mostly rely on worst case scenario and are not tight enough to explain the practical efficiency of diffusion models. On the other hand, numerical considerations mostly rely on Inception feature distributions through the FID metric [13].

Ideally, given a data distribution of interest, one would like to have an adapted estimation of the discrepancy between the data and the diffusion model samples, thus enabling adaptive hyperparameter selection for the sampling procedure. As a first step towards reaching this goal, in the present work we study diffusion models applied to Gaussian data distributions. While this setting has a priori no practical interest, since simulating Gaussian variates does not require a diffusion model, it provides a large parametric family of distributions for which the errors involved in diffusion model sampling can be completely understood.

When restricting the data distribution to be Gaussian, the resulting score function is a simple linear operator. Exploiting this specificity allows us to derive the following contributions **under the assumption that the data is Gaussian**:

- We give the exact solutions for both the backward SDE and probability flow ODE.
- We fully describe the Gaussian processes that occurs when using classical sampling discretization schemes.
- We derive exact 2-Wasserstein errors for the corresponding sample distributions and are able to assert for the influence of each error type on these errors, as illustrated by Figure 1.

Our theoretical study allows for a thorough experimental analysis of any numerical sampler, either stochastic or deterministic. In particular, it confirms the strength of best practice scheme such as Heun's method for the ODE flow [15].

While our theoretical analysis relies on an exact known score function, we conduct additional experiments to assess for the influence of the score approximation error. Surprisingly, in the context of texture synthesis, we show that with a score neural network trained for modeling a specific Gaussian micro-texture a stochastic Euler-Maruyama sampler is more faithful to the data distribution than Heun's method, thus highlighting the importance of the score approximation error in practical situations.

Plan of the paper: First, we recall in Section 2 the continuous framework for SDE-based diffusion models. Section 3 presents our main theoretical results detailing the exact backward SDE and probability flow ODE solutions when supposing the data distribution to be Gaussian. Section 4 gives explicit Wasserstein error formulas when sampling the corresponding processes, yielding to an ablation study for comparing the influence of each error type on several sampling schemes. Finally, in Section 5 we study numerically a special case of Gaussian distribution for texture synthesis in order to evaluate the influence of the score approximation error occurring with a standard network architecture.

2 Preliminaries: Score-based models through diffusion SDEs

This preliminary section follows the seminal work of Song *et al.* [26] and introduces specific notation to differentiate the exact backward process and the generative backward process obtained when starting from a white noise. Given a target distribution p_{data} over $\mathbb{R}^d$, the forward diffusion process is the following variance preserving SDE

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leq t \leq T, \quad \boldsymbol{x}_0 \sim p_{\mathrm{data}} \tag{1}$$

where $(\boldsymbol{w}_t)_{t\geq 0}$ is a d-dimension Brownian motion and β is a positive weight function. The distribution p_{data} is noised progressively and the function β is the variance of the added noise by time unit. We denote by p_t the density of $(\boldsymbol{x}_t)$ for $t > 0$ since p_{data} can be supported on a lower-dimensional manifold [5]. The SDE is designed so that p_T is close to the Gaussian standard distribution that we denote $\mathcal{N}_0$ in whole paper. Under some assumptions on the distribution p_{data} [21], the backward process $(\boldsymbol{x}_{T-t})_{0\leq t\leq T}$ verifies the backward SDE

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}} d\boldsymbol{w}_t, \quad 0 \leq t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{2}$$

The objective is now to solve this reverse equation to sample $\boldsymbol{y}_T \sim p_{\mathrm{data}}$. However, the distribution p_T is in general not known, and image[1] generation is achieved by sampling

$$d\tilde{\boldsymbol{y}}_t = \beta_{T-t}(\tilde{\boldsymbol{y}}_t + 2\nabla \log p_{T-t}(\tilde{\boldsymbol{y}}_t))dt + \sqrt{2\beta_{T-t}} d\boldsymbol{w}_t, \quad 0 \leq t < T, \quad \tilde{\boldsymbol{y}}_0 \sim \mathcal{N}_0. \tag{3}$$

Note that approximating p_T by $\mathcal{N}_0$ for the initialization $\boldsymbol{y}_0$ makes that the solution of the SDE (3) is not exactly the target distribution p_{data}. An alternative way to approximately sample p_{data} is to use that every diffusion process is associated with a deterministic process whose trajectories share the same marginal probability densities $(p_t)_{0<t\leq T}$ as the SDE [26]. The deterministic process associated with Equation (2) is

$$d\boldsymbol{x}_t = \left[-\beta_t \boldsymbol{x}_t - \beta_t \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right] dt, \quad 0 < t \leq T, \quad \boldsymbol{x}_0 \sim p_{\mathrm{data}}. \tag{4}$$

This ODE can be solved in reverse-time to sample $\boldsymbol{x}_0$ from $\boldsymbol{x}_T \sim p_T$. Given $(\boldsymbol{x}_t)_{0\leq t\leq T}$ solution of Equation (4), $(\boldsymbol{x}_{T-t})_{0\leq t\leq T}$ is solution of

$$d\boldsymbol{y}_t = \left[\beta_{T-t}\boldsymbol{y}_t + \beta_{T-t}\nabla_{\boldsymbol{y}} \log p_{T-t}(\boldsymbol{y}_t)\right] dt, \quad 0 \leq t < T. \tag{5}$$

Again, in practice, the ODE which is considered to achieve image generation is

$$d\widehat{\boldsymbol{y}}_t = \left[\beta_{T-t}\widehat{\boldsymbol{y}}_t + \beta_{T-t}\nabla_{\widehat{\boldsymbol{y}}} \log p_{T-t}(\widehat{\boldsymbol{y}}_t)\right] dt, \quad 0 \leq t < T, \quad \widehat{\boldsymbol{y}}_0 \sim \mathcal{N}_0, \tag{6}$$

where p_T is replaced by $\mathcal{N}_0$. As a consequence of this approximation, the property of conservation of the marginals $(p_t)_{0\leq t\leq T}$ does not occur. We denote by $(\tilde{q}_t)_{0\leq t\leq T}$, respectively $(\widehat{q}_t)_{0\leq t\leq T}$, the marginals of $(\tilde{\boldsymbol{y}}_t)_{0\leq t\leq T}$ and $(\widehat{\boldsymbol{y}}_t)_{0\leq t\leq T}$ and $\tilde{p}_t = \tilde{q}_{T-t}$,$\widehat{p}_t = \widehat{q}_{T-t}$ the marginals of $\left(\tilde{\boldsymbol{y}}_{T-t}\right)_{0\leq t\leq T}$ and $\left(\widehat{\boldsymbol{y}}_{T-t}\right)_{0\leq t\leq T}$ such that $\tilde{p}_t$ and $\widehat{p}_t$ are approximations of p_t.

[1] Although we may refer to data as images, our analysis is fully general and applies to any vector-valued diffusion model.

3 Exact SDE and ODE solutions

Our approach relies on deriving explicit solutions to the various SDE and ODE. We begin with the forward SDE in full generality obtained in applying the variation of constants (see the proof in Appendix B.1). This resolution also provides an ODE verified by the covariance matrix of $\boldsymbol{x}_t$, that we denote $\boldsymbol{\Sigma}_t = \mathrm{Cov}(\boldsymbol{x}_t)$.

Proposition 1 (Solution of the forward SDE)**.** *The strong solution of Equation* (1) *can be written as:*

$$\boldsymbol{x}_t = e^{-B_t}\boldsymbol{x}_0 + \boldsymbol{\eta}_t, \quad 0 \leq t \leq T, \tag{7}$$

where $B_t = \int_0^t \beta_s ds$ *and* $\boldsymbol{\eta}_t = e^{-B_t}\int_0^t e^{B_s}\sqrt{2\beta_s}d\boldsymbol{w}_s$ *is a Gaussian process independent of* $\boldsymbol{x}_0$ *whose covariance matrix is* $(1 - e^{-2B_t})\boldsymbol{I}$*. Consequently, the covariance matrix* $\boldsymbol{\Sigma}_t$ *of* $\boldsymbol{x}_t$ *is*

$$\boldsymbol{\Sigma}_t = e^{-2B_t}\boldsymbol{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}. \tag{8}$$

where $\boldsymbol{\Sigma}$ *is the covariance matrix of* $\boldsymbol{x}_0 \sim p_{\mathrm{data}}$*. Futhermore,* $\boldsymbol{\Sigma}_t$ *is invertible for* $t > 0$ *and verifies the matrix-valued ODE*

$$d\boldsymbol{\Sigma}_t = 2\beta_t(\boldsymbol{I} - \boldsymbol{\Sigma}_t)dt, \quad 0 < t \leq T. \tag{9}$$

For a general data distribution p_{data}, solving the backward SDE in infeasible, the main reason being that the expression of the score function to integrate is unknown. To circumvent this obstacle, we now suppose that the data distribution is Gaussian.

Assumption 1 (Gaussian assumption)**.** p_{data} ***is a centered Gaussian distribution*** $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$**.**

Note that $\boldsymbol{\Sigma}$ may be non-invertible and thus p_{data} supported on a strict subspace of $\mathbb{R}^d$, a special case of manifold hypothesis. Consequently, the matrix $\boldsymbol{\Sigma}_t$ is in general only invertible for $t > 0$. Under Gaussian assumption, $(\boldsymbol{x}_t)$ is a Gaussian process with marginal distribution $p_t = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_t)$ and consequently the score is a the linear function

$$\nabla \log p_t(\boldsymbol{x}) = -\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}, \quad 0 < t \leq T. \tag{10}$$

Note that the linearity of the diffusion score characterizes Gaussian distributions as detailed by Proposition 5 in Appendix A.

The cornerstone of our work is that under Gaussian assumption we can derive an exact solution of the backward SDE, without supposing that the initial condition is Gaussian.

Proposition 2 (Solution of the backward SDE under Gaussian assumption)**.** *Under Gaussian assumption, the strong solution to Equation* (2) *can be written as:*

$$\boldsymbol{y}_t = e^{-(B_T - B_{T-t})}\boldsymbol{\Sigma}_{T-t}\boldsymbol{\Sigma}_T^{-1}\boldsymbol{y}_0 + \boldsymbol{\xi}_t, \quad 0 \leq t \leq T \tag{11}$$

where $\boldsymbol{\xi}_t = e^{-(B_T - B_{T-t})}\boldsymbol{\Sigma}_{T-t}\int_0^t \boldsymbol{\Sigma}_{T-s}^{-1}e^{-(B_T - B_{T-s})}\sqrt{2\beta_{T-s}}d\boldsymbol{w}_s$ *is a Gaussian process with covariance matrix* $\mathrm{Cov}(\boldsymbol{\xi}_t) = \boldsymbol{\Sigma}_{T-t} - e^{-2(B_T - B_{T-t})}\boldsymbol{\Sigma}_{T-t}^2\boldsymbol{\Sigma}_T^{-1}$*. Finally:*

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})}\boldsymbol{\Sigma}_{T-t}^2\boldsymbol{\Sigma}_T^{-1}\left(\boldsymbol{\Sigma}_{T-t}^{-1}\,\mathrm{Cov}(\boldsymbol{y}_0)\boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_{T-t} - \boldsymbol{I}\right), \tag{12}$$

and in particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ *and* $\boldsymbol{\Sigma}$ *commute,*

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})}\boldsymbol{\Sigma}_{T-t}^2\boldsymbol{\Sigma}_T^{-1}\left[\boldsymbol{\Sigma}_T^{-1}\,\mathrm{Cov}(\boldsymbol{y}_0) - \boldsymbol{I}\right] \tag{13}$$

While not as straightforward as the forward case, the proof also relies on applying the variation of constants and is given in Appendix B.2. As shown by the following proposition (proved in Appendix B.3), the flow ODE also has an explicit solution under Gaussian assumption which related to optimal transport.

Proposition 3 (Solution of the ODE probability flow under Gaussian assumption)**.** *The solution to the probability flow ODE* (4) *under Gaussian assumption corresponds to the optimal transport map between* p_T *and* p_{data}*. More precisely, for any* $\boldsymbol{y}_0$*,*

$$\boldsymbol{y}_t = \boldsymbol{\Sigma}_T^{-1/2}\boldsymbol{\Sigma}_{T-t}^{1/2}\boldsymbol{y}_0, \quad 0 \leq t \leq T,$$

is the solution of the reverse-time ODE (5)*. Consequently, the covariance matrix* $\mathrm{Cov}(\boldsymbol{y}_t)$ *verifies*

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1/2}\boldsymbol{\Sigma}_{T-t}^{1/2}\,\mathrm{Cov}(\boldsymbol{y}_0)\boldsymbol{\Sigma}_{T-t}^{1/2}\boldsymbol{\Sigma}_T^{-1/2}, \quad 0 \leq t \leq T, \tag{14}$$

and in particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ *and* $\boldsymbol{\Sigma}$ *commute,*

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_{T-t}\,\mathrm{Cov}(\boldsymbol{y}_0), \quad 0 \leq t \leq T. \tag{15}$$

Here we must highlight a subtle issue: Whatever the initial distribution is, the ODE solution consists in applying the optimal transport map between p_T and p_{data}. Since in practice one cannot truly sample p_T, the resulting flow is not an optimal transport flow and the distribution of $\boldsymbol{y}_T$ differs from p_{data}.

Links with related work. Some parts of the previous propositions have been stated in previous work. The expression of the SDE solution of Proposition 1 is given without proof in [12] and the ODE verified by the variance is given in [26] (citing [22]) but it is generalized here to the full covariance matrix (Equation (9)). To the best of our knowledge, Proposition 2 is new and one the most important contribution of the paper. Similar to our approach, Gaussian mixtures have been studied in the context of diffusion model [29, 30, 23] since they also provide an explicit analytical score. However, solving the backward SDE is not feasible for Gaussian mixtures as far as we know. The ODE [17, 16] can be interpreted in the infinite time as an optimal transport (OT) between the prior distribution and the Gaussian standard distribution. The relation between optimal transport and probability flow ODE (also called Fokker-Planck ODE) has been discussed in [17, 16] in the asymptotic case where $T \mapsto +\infty$. Both these papers discussed the Gaussian case, but our Proposition 3 highlights that the generated process is not an optimal transport flow due to the intialiazation error.

4 Exact Wasserstein errors

The specificity of the Gaussian case allows us to study precisely the different types of error with the expression of the explicit solution of the backward SDE. In what follows, we designate by Wasserstein distance the 2-Wasserstein distance which is known in closed forms when applied to Gaussian distributions[8]. For two centered Gaussians $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2)$ such that $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ are simultaneously diagonalizable with respective eigenvalues $(\lambda_{i,1})_{1\leq i\leq d}, (\lambda_{i,2})_{1\leq i\leq d}$,

$$\mathbf{W}_2(\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1), \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2))^2 = \sum_{1\leq i\leq d} (\sqrt{\lambda_{i,1}} - \sqrt{\lambda_{i,2}})^2. \tag{16}$$

In the literature, the quality of the diffusion models is measured with FID [13] which is the $\mathbf{W}_2$-error between Gaussians fitted to the Inception features [27] of two discrete datasets. Here we use the $\mathbf{W}_2$-errors directly in data space, which is more informative and allows us to provide theoretical $\mathbf{W}_2$-errors. To illustrate our theoretical results, we consider the CIFAR-10 Gaussian distribution, that is, the Gaussian distribution such that $\boldsymbol{\Sigma}$ is the empirical covariance of the CIFAR-10 dataset. As shown in Appendix C, images produced by this model are not interesting due to a lack of structure, but the corresponding covariance has the advantage of reflecting the complexity of real data.

The initialization error. As discussed in Sections 2 and 3, the marginals of both generative processes $\tilde{\boldsymbol{y}}$ and $\widehat{\boldsymbol{y}}$ following respectively Equation (6) and Equation (3) slightly differs from p_t due to their common white noise initial condition. This implies an error that we call the initialization error. The distance between $(\tilde{p}_t)_{0\leq t\leq T}$, $(\tilde{p}_t)_{0\leq t\leq T}$ and $(p_t)_{0\leq t\leq T}$ can be explicitly studied in the Gaussian case with the following proposition (proved in Appendix B.4).

Proposition 4 (Marginals of the generative processes under Gaussian assumption)**.** *Under Gaussian assumption, $(\tilde{\boldsymbol{y}}_t)_{0\leq t\leq T}$ and $(\widehat{\boldsymbol{y}}_t)_{0\leq t\leq T}$ are Gaussian processes. At each time t, $\tilde{p}_t$ is the Gaussian distribution $\mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}_t)$ with $\tilde{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_t + e^{-2(B_T - B_t)}\boldsymbol{\Sigma}_t^2\boldsymbol{\Sigma}_T^{-1}(\boldsymbol{\Sigma}_T^{-1} - \boldsymbol{I})$ and $\widehat{p}_t$ is the Gaussian distribution $\mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}}_t)$ with $\widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_t$. For all $0 \leq t \leq T$, the three covariance matrices $\boldsymbol{\Sigma}_t$, $\tilde{\boldsymbol{\Sigma}}_t$ and $\widehat{\boldsymbol{\Sigma}}_t$ share the same range. Furthermore, for all $0 \leq t \leq T$,*

$$\boldsymbol{W}_2(\tilde{p}_t, p_t) \leq \boldsymbol{W}_2(\widehat{p}_t, p_t) \tag{17}$$

which shows for $t = 0$ that the SDE sampler is a better sampler than the ODE sampler when the exact score is konwn.

In practice the initialization error for the SDE and ODE samplers may vary by several orders of magnitude, as shown for the CIFAR-10 example in Figure 1.(a) (solid lines).

The discretization error. The implementation of the SDE and the ODE implies to choose a discrete numerical scheme. We propose to study four different schemes presented in Table 1. The classical

SDE schemes	Euler-Maruyama (EM)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EM}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} + \Delta_t \beta_{T-t_k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_k}} \boldsymbol{z}_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0 \end{cases}$	(19)
	Exponential integrator (EI)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EI}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EI}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} + \gamma_{1,k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} \right) + \sqrt{2\gamma_{2,k}} z_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0 \end{cases}$ where $\gamma_{1,k} = \exp(B_{T-t_k} - B_{T-t_{k+1}}) - 1$ and $\gamma_{2,k} = \frac{1}{2}(\exp(2B_{T-t_k} - 2B_{T-t_{k+1}}) - 1)$	(20)
ODE schemes	Explicit Euler	$\begin{cases} \widehat{\boldsymbol{y}}_0^{\Delta,\text{Euler}} & \sim \mathcal{N}_0 \\ \widehat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Euler}} & = \widehat{\boldsymbol{y}}_k^{\Delta,\text{Euler}} + \Delta_t f(t_k, \widehat{\boldsymbol{y}}_k^{\Delta,\text{Euler}}) \quad \text{with } f(t, \boldsymbol{y}) = \beta_{T-t}\boldsymbol{y} - \beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y} \end{cases}$	(21)
	Heun's method	$\begin{cases} \widehat{\boldsymbol{y}}_0^{\Delta,\text{Heun}} & \sim \mathcal{N}_0 \\ \widehat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}} & = \widehat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \Delta_t f(t_k, \widehat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) \quad \text{with } f(t, \boldsymbol{y}) = \beta_{T-t}\boldsymbol{y} - \beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y} \\ \widehat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Heun}} & = \widehat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \frac{\Delta_t}{2} \left(f(t_k, \widehat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) + f(t_{k+1}, \widehat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}}) \right) \end{cases}$	(22)

Table 1: **Stochastic and deterministic discretization schemes**. EM and EI disctretize the backward SDE (3), Euler and Heun schemes discretize of the probability flow ODE (6).

Euler-Maruyama (EM) is used in [26] and the exponential integrator (EI) in [5] to sample from the SDE (3). The Euler method is the simplest ODE solver and Heun's scheme is recommended in [15] to model the ODE (6). Under Gaussian assumption, the eigenvalues of the covariance matrix can be computed numerically recursively for each scheme to evaluate the Wasserstein distance. More precisely, all the covariance matrices are diagonalizable in the diagonalization basis of $\boldsymbol{\Sigma}$, and thus Equation (16) is valid. For example, denoting $(\lambda_i^t)_{1\leq i\leq d}$ the eigenvalues of $\boldsymbol{\Sigma}_t$ and $\left(\lambda_i^{EM,k}\right)_{1\leq i\leq d}$ the eigenvalues of the covariance matrix of the Euler-Maruyama discretization of the SDE at the kth step, $1 \leq k \leq N-1$, the relation verified by these eigenvalues is

$$\lambda_i^{\text{EM},k+1} = \left(1 + \Delta_t \beta_{T-t_k}(1 - \frac{2}{\lambda_i^{T-t_k}})\right)^2 \lambda_i^{\text{EM},k} + 2\Delta_t \beta_{T-t_k}, 1 \leq i \leq d, 0 \leq k \leq N-2 \quad (18)$$

with initialization $\lambda_i^{0,\text{EM}} = 1, 1 \leq i \leq d$. For each scheme, we recursively compute the eigenvalues at each time discretization and present the observed Wasserstein distance in Figure 1.(a). We can observe that Heun's method provide the lower Wasserstein distance, followed by EM, EI and the Euler scheme. Note that the discrete schemes does not preserve the range of the covariance matrix, contrary to the continuous formulas. This explains the fact that the Wasserstein distance increases at the final step.

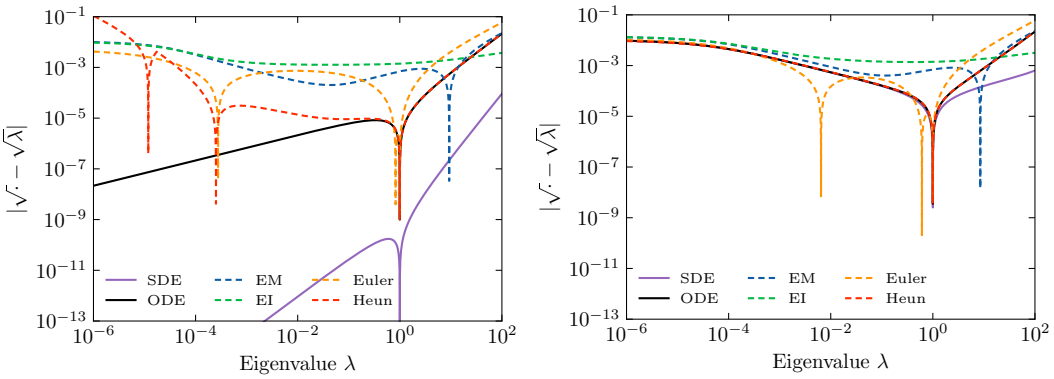
The truncation error. As discussed in [26], it is preferable to study the backward process on $[\varepsilon, T]$ instead of $[0, T]$ because the score is a priori not defined for $t = 0$, which occurs in our case if $\boldsymbol{\Sigma}$ is not invertible. This approximation is called the truncation error. As a consequence, even without error initialization, the backward process leads to p_ε and not p_0. Under Gaussian assumption, it is possible to explicit this error with the expression given in Proposition 3 and 2 as done in Figure 5.(b) for both continuous and numerical solutions. For the standard practice truncation time $\varepsilon = 10^{-3}$ [26, 15], all numerical schemes have an error close to the corresponding continuous solution. Using a lower ε value is only relevant for the continuous SDE solution.

Ablation study. We propose in Table 2 an ablation study to monitor the magnitude of each error and their accumulation for various sampling schemes for the CIFAR-10 example. In accordance with Proposition 4, the initialization error influences the ODE schemes, while SDE schemes are not affected. Schemes having a sufficient number of steps are not sensitive to the truncation error for $\varepsilon < 10^{-2}$. The discretization error is the more important approximation but it becomes very low for a sufficient number of steps. The lower Wasserstein error is provided by Heun's method with 1000 steps, $\varepsilon = 10^{-5}$. As [15], our conclusions lead to the choice of Heun's scheme as the go-to method.

Influence of eigenvalues. The above observations and conclusions are observed on the CIFAR-10 Gaussian. However, in general, they depend on the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$. Indeed, as seen in Equation (16), the Wasserstein distance is separable and each eigenvalue contributes to increase it. In Figure 2, we evaluate the contribution of each eigenvalue by plotting $\lambda \mapsto |\sqrt{\lambda} - \sqrt{\lambda^{\text{scheme}}}|$ for each scheme. Figure 2.(a) demonstrates that for the continuous equations, the error increases with the eigenvalues except for a strong decrease for $\lambda = 1$. Besides, as proved in the proof of Proposition 4 (see Appendix B.4), the error for the SDE is always lower than the error for the ODE.

		Continuous		$N = 50$		$N = 250$		$N = 500$		$N = 1000$	
		p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$
EM	$\varepsilon = 0$	0	6.7E-4	4.77	4.77	0.65	0.65	0.31	0.31	0.15	0.16
	$\varepsilon = 10^{-5}$	2.5E-3	2.6E-3	4.77	4.77	0.65	0.65	0.31	0.31	0.16	0.16
	$\varepsilon = 10^{-3}$	0.17	0.17	4.67	4.67	0.69	0.69	0.39	0.39	0.27	0.27
	$\varepsilon = 10^{-2}$	1.35	1.35	4.56	4.56	1.69	1.69	1.50	1.50	1.42	1.42
EI	$\varepsilon = 0$	0	6.7E-4	2.81	2.81	0.57	0.57	0.30	0.30	0.16	0.16
	$\varepsilon = 10^{-5}$	2.5E-3	2.6E-3	2.81	2.81	0.57	0.57	0.30	0.30	0.16	0.16
	$\varepsilon = 10^{-3}$	0.17	0.17	2.91	2.91	0.66	0.66	0.41	0.41	0.28	0.28
	$\varepsilon = 10^{-2}$	1.35	1.35	3.93	3.93	1.76	1.76	1.55	1.55	1.45	1.45
Euler	$\varepsilon = 0$	0	0.07	1.72	1.78	0.38	0.44	0.19	0.26	0.10	0.17
	$\varepsilon = 10^{-5}$	2.5E-3	0.07	1.72	1.78	0.38	0.44	0.20	0.26	0.10	0.17
	$\varepsilon = 10^{-3}$	0.17	0.19	1.72	1.78	0.42	0.48	0.27	0.32	0.21	0.25
	$\varepsilon = 10^{-2}$	1.35	1.36	2.21	2.25	1.41	1.43	1.37	1.38	1.36	1.37
Heun	$\varepsilon = 0$	0	0.07	7.09	7.09	0.72	0.73	0.21	0.22	0.05	0.09
	$\varepsilon = 10^{-5}$	2.5E-3	0.07	6.48	6.48	0.64	0.65	0.18	0.20	0.05	0.09
	$\varepsilon = 10^{-3}$	0.17	0.19	0.56	0.57	0.13	0.15	0.16	0.18	0.17	0.19
	$\varepsilon = 10^{-2}$	1.35	1.36	1.37	1.38	1.35	1.36	1.35	1.36	1.35	1.36

Table 2: **Ablation study of Wasserstein errors for the CIFAR-10 Gaussian.** For a given discretization scheme, the table presents the Wasserstein distance associated with the truncation error for different values of ε. The columns p_T and $\mathcal{N}_0$ show the influence of the initialization error. The continuous column corresponds to the continuous SDE or ODE linked with the scheme (identical values for EM, EI and Euler, Heun).



(a) Initialization error at final time

(b) Truncation error at final time for $\varepsilon = 10^{-3}$

Figure 2: **Eigenvalue contribution to the Wasserstein error.** The magnitude of the Wasserstein error is influenced by the eigenvalues of the covariance of the Gaussian distribution. Left: Contribution to the Wasserstein error for the continuous equations and the discretization schemes with standard initialization $\mathcal{N}_0$. Right: Same plot when using a truncation time $\varepsilon = 10^{-3}$. All schemes use $N = 1000$ steps. While we prove that the continuous SDE is always better than the continuous ODE (Proposition 4), it is not the same for the discrete schemes. With a truncation time $\varepsilon = 10^{-3}$ (b), Heun's method is nearly as good as the continuous ODE solution for all eigenvalues, which shows it is well-adapted to any Gaussian distribution.

Unfortunately, once discretized the stochastic schemes are not as good as the continuous solutions. The EI scheme is the more stable along the range of eigenvalues but in the end it is in general more costly than the others in terms of Wasserstein error. Without truncation time, Heun's method fails for low eigenvalues because $\mathbf{\Sigma}$ is not stably invertible. However, as seen in Figure 2.(b), with a truncation time $\varepsilon = 10^{-3}$, Heun's method is very close to the continuous ODE solution. This shows that for any Gaussian distribution Heun's method introduces nearly no additional discretization error, making this scheme the one to favor in practice.

5 Numerical study of the score approximation

So far our theoretical and numerical study has been conducted under the hypothesis that the score function is known, thus discarding the evaluation of the score approximation. In practice, for general data distribution, the score function is parameterized by a neural network trained using denoising score-matching. This learned score function is not perfect and while theoretical studies assume the network to be close to the theoretical one (with uniform or adaptative bounds, see the discussion in [5]), such an hypothesis is hard to check in practice, especially in our non compact setting. Thus, we propose in this section to train a diffusion models on a Gaussian distribution and evaluate numerically the impact of the score approximation.

The Gaussian ADSN distribution for microtextures. So far our running example was the CIFAR-10 Gaussian but we will now turn to another example that produces visually interesting images, namely Gaussian micro-textures. We consider the asymptotic discrete spot noise (ADSN) distribution [11] associated with an RGB texture $\boldsymbol{u} \in \mathbb{R}^{3\times M\times N}$ which is defined as the stationary Gaussian distribution that has covariance equal the autocorrelation of $\boldsymbol{u}$. More precisely, this distribution is sampled using convolution with a white Gaussian noise [11]: Denoting $m \in \mathbb{R}^3$ the channelwise mean of $\boldsymbol{u}$ and $\mathbf{t}_c = \frac{1}{\sqrt{MN}}(\boldsymbol{u}_c - m_c)$, $1 \leq c \leq 3$, its associated *texton*, for $\boldsymbol{w} \sim \mathcal{N}_0$ of size $M \times N$ the channelwise convolution $\boldsymbol{x} = m + \mathbf{t} \star \boldsymbol{w} \in \mathbb{R}^{3\times M\times N}$ follows $\mathrm{ADSN}(\boldsymbol{u})$. This distribution is the Gaussian $\mathcal{N}(m, \mathbf{\Sigma})$. To deal with zero mean Gaussian, adding the mean m is considered as a post-processing to visualize samples and we study $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. The matrix $\mathbf{\Sigma}$ is a well-known convolution matrix [9], its eigenvectors and associated eigenvalues can be computed in the Fourier domain, as done in Appendix E.2. $\mathbf{\Sigma}$ admits the eigenvalues $\lambda_1^{\xi,\mathrm{ADSN}} = |\widehat{\mathbf{t}}_1|^2(\xi) + |\widehat{\mathbf{t}}_2|^2(\xi) + |\widehat{\mathbf{t}}_3|^2(\xi), \xi \in \mathbb{R}^{M\times N}$ and 0 with multiplicity $2MN$ and we can conduct the same analysis as before (see Appendix D). To evaluate if a set of N_{samples} sampled images is close to the ADSN distribution p_{data}, we evaluate a problem-specific empirical Wasserstein distance: Supposing that the N_{samples} are drawn from a Gaussian distribution $p^{\text{emp.}} = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ such that $\mathbf{\Gamma}$ admits the same eigenvectors as $\mathbf{\Sigma}$, we compute

$$\mathbf{W}_2^{\text{emp.}}(p^{\text{emp.}}, p_{\mathrm{data}}) = \sqrt{\sum_{\xi\in\mathbb{R}^{3M\times N}} \left(\sqrt{\lambda_1^{\xi,\text{emp.}}} - \sqrt{\lambda_1^{\xi,\mathrm{ADSN}}}\right)^2 + \lambda_2^{\xi,\text{emp.}} + \lambda_3^{\xi,\text{emp.}}} \tag{23}$$

where $(\lambda_i^{\xi,\text{emp.}})_{\xi\in\mathbb{R}^{M\times N}, 1\leq i\leq 3}$ are estimators of the eigenvalues of $\mathbf{\Gamma}$ given in Appendix E.3.

Learning the score function. We train the network using the code[2] associated with the paper [26]. We choose the architecture of DDPM, which is a U-Net described in [14], with the parameters proposed for the dataset CelebaHQ256 to deal with the 256×256 ADSN model associated with the top-left image of Figure 3. We use the training procedure corresponding to DDPM cont. in [26]. β is linear from 0.05 to 10 with $T = 1$. We train over 1.3M iterations, and we generate at each iteration a new batch of ADSN samples. We implement the stochastic EM and derministic Heun schemes replacing the exact score by its learned version with $N = 1000$ steps and a trunction time $\varepsilon = 10^{-3}$. We name p_θ^{EM} and p_θ^{Heun}, the corresponding distributions and present samples in Figure 3. Both distributions accumulate the four error types.

Evaluation of the score approximation. It is not possible to compute theoretically the Wasserstein distance between $p_{\mathrm{data}} = \mathrm{ADSN}(\boldsymbol{u})$ and $p_\theta^{\text{EM}}, p_\theta^{\text{Heun}}$ due to the non-linearity of the learned score. To compute an empirical Wasserstein error between it, we use Equation (23). Let us precise that this approximation underestimates the real Wasserstein distance since it wrongly assumes that the distributions p_θ^{EM}, p_θ^{Heun} are Gaussian with a covariance matrix diagonalizable in the same basis than the covariance matrix $\mathbf{\Sigma}$ of $\mathrm{ADSN}(\boldsymbol{u})$. We complete this dedicated empirical measure with the standard FID. This metrics are reported in Table 3 where for theoretical distributions that are fast to sample we add the standard deviations computed on 25 different $50k$-samplings. For this Gaussian distribution, the score approximation is by far the most impactful source of error. We observe that the stochastic EM sampling is more resilient to score approximation than the deterministic Heun's scheme, resulting in out-of-distribution samples (Figure 3). We may explain this behavior by recalling the results of Proposition 4 that shows that SDE solutions are less sensitive to initialization errors than ODE. Indeed, adding noise at each iteration tends to mitigate the accumulated errors, and score approximation may be consider as some initialization error ocuring at each step.

[2]Code available at `https://github.com/yang-song/score_sde_pytorch`

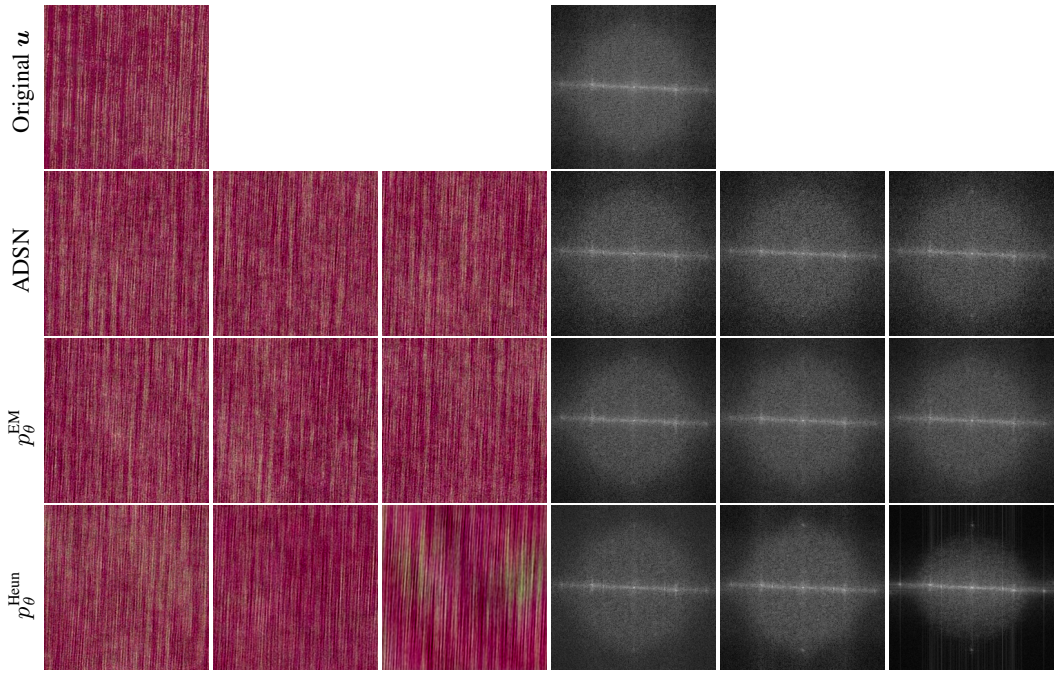


Figure 3: **Texture samples generated with the learned score.** First row: original image $\boldsymbol{u}$ and its DFT modulus (for all DFT modulus we display the sum of the DFT modulus of the three color channels and apply a logarithmic contrast change). Second row: three samples of $\mathrm{ADSN}(\boldsymbol{u})$ with their associated DFT moduli. Third and fourth row: Samples generated with the learned score with EM and Heun's discretization schemes and their associated DFT moduli. While both schemes use the same learned score function, the generation with Heun's scheme can produce out-of-distribution samples, as seen with the third sample.

	Exact score distribution			Learned score distribution	
p	$\mathbf{W}_2(p,p_{\text{data}})\downarrow$	$\mathbf{W}_2^{\text{emp.}}(p^{\text{emp.}},p_{\text{data}})\downarrow$	$\text{FID}(p^{\text{emp.}},p_{\text{data}}^{\text{emp.}})\downarrow$	$\mathbf{W}_2^{\text{emp.}}(p_\theta^{\text{emp.}},p_{\text{data}}^{\text{emp.}})\downarrow$	$\text{FID}(p_\theta^{\text{emp.}},p_{\text{data}}^{\text{emp.}})\downarrow$
EM	5.16	5.1630±7E-5	0.0891±8E-4	15.6	1.02
Heun	3.73	3.7323±2E-4	0.0447±6E-4	56.7	19.4

Table 3: **Numerical evaluation of the score approximation for a Gaussian microtexture model.** For two schemes, the EM discretization of the backward SDE and Heun's method associated with the flow ODE, the table shows the Wasserstein distance and FID for theoretical and learned distributions. The theoretical $\mathbf{W}_2$ value is computed with explicit formulas, as done in Table 4. The FID and empirical $\mathbf{W}_2$ w.r.t the theoretical distribution are computed on 25 samplings of 50k images while only one sampling of 50k images is drawn for the parametric distributions (to limit computation time).

6 Conclusion

By restricting the analysis of diffusion models to the specific case of Gaussian distributions, we were able to derive exact solutions for both the backward SDE and its associated probability flow ODE. Additionally, we characterized the discrete Gaussian processes arising when discretizing these equations. This allowed us to provide exact Wasserstein errors for the initialization error, the discretization error, and the truncation error as well as any of their combinations. This theoretical analysis led to conclude that Heun's scheme is the best numerical solution, in accordance with empirical previous work [15].

To conclude our work we conducted an empirical analysis with a learned score function using standard architecture which showed that the score approximation error may be the most important one in practice. This suggests that assessing the quality of learned score functions is an important research direction for future work.

Acknowledgements: The authors acknowledge the support of the project MISTIC (ANR-19-CE40-005).

References

[1] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4672–4712. PMLR, 23–29 Jul 2023.

[3] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ODE is provably fast. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[4] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023.

[5] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*, 2022.

[6] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 17695–17709. Curran Associates, Inc., 2021.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

[8] D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982.

[9] Sira Ferradans, Gui-Song Xia, Gabriel Peyré, and Jean-François Aujol. Static and dynamic texture mixing using optimal transport. In *Scale Space and Variational Methods in Computer Vision*, 2013.

[10] Giulio Franzese, Simone Rossi, Lixuan Yang, Alessandro Finamore, Dario Rossi, Maurizio Filippone, and Pietro Michiardi. How much is enough? a study on diffusion times in score-based generative models. *Entropy*, 25(4), 2023.

[11] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, 20(1):257–267, 2011.

[12] Xuefeng Gao and Lingjiong Zhu. Convergence analysis for general probability flow odes of diffusion models in wasserstein distances. *ArXiv*, abs/2401.17958, 2024.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

[15] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

[16] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding DDPM latent codes through optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] Hugo Lavenant and Filippo Santambrogio. The flow map of the fokker–planck equation does not provide optimal transport. *Applied Mathematics Letters*, 133:108225, 2022.

[18] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[19] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[20] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer Berlin Heidelberg, 2010.

[21] E. Pardoux. Grossissement d'une filtration et retournement du temps d'une diffusion. In Jacques Azéma and Marc Yor, editors, *Séminaire de Probabilités XX 1984/85*, pages 48–55, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.

[22] S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[23] Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the DDPM objective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[25] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[26] 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.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] Li Kevin Wenliang and Ben Moran. Score-based generative model learn manifold-like structures with constrained mixing. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[29] Martin Zach, Erich Kobler, Antonin Chambolle, and Thomas Pock. Product of gaussian mixture diffusion models. *Journal of Mathematical Imaging and Vision*, Mar 2024.

[30] Martin Zach, Thomas Pock, Erich Kobler, and Antonin Chambolle. Explicit diffusion of gaussian mixture model based image priors. In Luca Calatroni, Marco Donatelli, Serena Morigi, Marco Prato, and Matteo Santacesaria, editors, *Scale Space and Variational Methods in Computer Vision*, pages 3–15, Cham, 2023. Springer International Publishing.

A Characterization of Gaussian distributions through diffusion models

The following proposition shows that our Gaussian assumption occurs if and only if the score function is linear.

Proposition 5. *The three following propositions are equivalent:*

(i) $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ *for some covariance* $\boldsymbol{\Sigma}$*.*

(ii) $\forall t > 0, \nabla_x \log p_t(\boldsymbol{x})$ *is linear w.r.t* $\boldsymbol{x}$*.*

(iii) $\exists t > 0, \nabla_x \log p_t(\boldsymbol{x})$ *is linear w.r.t* $\boldsymbol{x}$*.*

In this case, for $t > 0$*,* $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = -\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}$*, with* $\boldsymbol{\Sigma}_t$ *defined in Proposition 1.*

Proof. $(ii) \Rightarrow (iii)$ is clear.

If (i), for $t > 0$, $p_t(\boldsymbol{x}) = C_t \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}\right)$. Consequently, $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = -\boldsymbol{\Sigma}_t^{-1}\boldsymbol{x}$ and $(i) \Rightarrow (ii)$

If (iii), there exists A such that $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = A\boldsymbol{x}$. Consequently, $p_t(\boldsymbol{x}) = C_t \exp(-\frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x})$ and $\boldsymbol{x}_t$ is Gaussian. This provides that $\boldsymbol{x}_0 = e^{B_t}\boldsymbol{x}_t - \boldsymbol{\eta}_t$ is Gaussian and $(iii) \Rightarrow (i)$. □

B Proofs of Section 3

B.1 Proposition 1: Solution of the forward SDE

We aim at solving:

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad \boldsymbol{x}_0 \sim p_{\text{data}}. \tag{24}$$

By considering $\boldsymbol{z}_t = e^{B_t}\boldsymbol{x}_t$ where $B_t = \int_0^t \beta_s ds$,

$$d\boldsymbol{z}_t = \beta_t e^{B_t}\boldsymbol{x}_t + e^{B_t} d\boldsymbol{x}_t = \beta_t e^{B_t}\boldsymbol{x}_t + e^{B_t}(-\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t) = \sqrt{2\beta_t} e^{B_t} d\boldsymbol{w}_t. \tag{25}$$

Consequently, for $0 \leq t \leq T$,

$$\boldsymbol{z}_t = \boldsymbol{z}_0 + \int_0^t \sqrt{2\beta_s} e^{B_s} d\boldsymbol{w}_s, \boldsymbol{z}_0 = e^{B_0}\boldsymbol{x}_0 = \boldsymbol{x}_0 \tag{26}$$

and for $0 \leq t \leq T$,

$$\boldsymbol{x}_t = e^{-B_t}\boldsymbol{z}_t = e^{-B_t}\boldsymbol{x}_0 + e^{-B_t}\int_0^t e^{B_s}\sqrt{2\beta_s} d\boldsymbol{w}_s = e^{-B_t}\boldsymbol{x}_0 + \boldsymbol{\eta}_t. \tag{27}$$

By Itô's isometry (see e.g [20]),

$$\text{Var}\left(\int_0^t e^{B_s}\sqrt{2\beta_s} d\boldsymbol{w}_s\right) = \int_0^t 2\beta_s e^{2B_s} ds = [e^{2B_s}]_0^t = e^{2B_t} - e^{2B_0} = e^{2B_t} - 1 \tag{28}$$

which provides the covariance matrix of $\boldsymbol{\eta}_t$:

$$\text{Cov}\left(\boldsymbol{\eta}_t\right) = e^{-2B_t}(e^{2B_t} - 1)\boldsymbol{I} = \left(1 - e^{-2B_t}\right)\boldsymbol{I}. \tag{29}$$

Because $\boldsymbol{x}_0$ and $\boldsymbol{\eta}_t$ are independent, $\boldsymbol{\Sigma}_t = e^{-2B_t}\boldsymbol{\Sigma} + \left(1 - e^{-2B_t}\right)\boldsymbol{I}$.

And,

$$d\boldsymbol{\Sigma}_t = -2\beta_t e^{-2B_t}(\boldsymbol{\Sigma} - \boldsymbol{I})dt = \left[-2\beta_t \boldsymbol{\Sigma}_t + 2\beta_t \boldsymbol{I}\right] dt = -(2\beta_t \boldsymbol{\Sigma}_t - 2\beta_t \boldsymbol{\Sigma}_T^{-1})\boldsymbol{\Sigma}_t dt. \tag{30}$$

B.2 Proposition 2: Solution of the backward SDE under Gaussian assumption

We aim at solving

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t - 2\nabla \boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y}_t)dt + \sqrt{2\beta_{T-t}} d\boldsymbol{w}_t, \quad 0 \leq t \leq T \tag{31}$$

Denoting $C_t = \int_0^t \beta_{T-s} ds$, by considering $\boldsymbol{z}_t = \boldsymbol{\Sigma}_{T-t}^{-1} e^{C_t}\boldsymbol{y}_t$,

$$\begin{aligned}
d\boldsymbol{z}_t &= e^{C_t}\boldsymbol{\Sigma}_{T-t}^{-1} d\boldsymbol{y}_t - e^{C_t} d[\boldsymbol{\Sigma}^{-1}]_{T-t}\boldsymbol{y}_t dt + \beta_{T-t}\boldsymbol{z}_t dt && (32)\\
&= \left[\boldsymbol{\Sigma}_{T-t}^{-1} e^{C_t}\beta_{T-t}(\boldsymbol{y}_t - 2\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y}_t) - \beta_{T-t}\boldsymbol{z}_t + 2\beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{z}_t\right] dt + \sqrt{2\beta_{T-t}} e^{C_t}\boldsymbol{\Sigma}_{T-t}^{-1} d\boldsymbol{w}_t && (33)\\
&\text{(using Equation (9))} && (34)\\
&= \beta_{T-t}(1 - 2\boldsymbol{\Sigma}_{T-t}^{-1})\boldsymbol{z}_t dt - \beta_{T-t}\boldsymbol{z}_t dt + 2\beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{z}_t dt + e^{C_t}\sqrt{2\beta_{T-t}}\boldsymbol{\Sigma}_{T-t}^{-1} d\boldsymbol{w}_t && (35)\\
&= \sqrt{2\beta_{T-t}} e^{C_t}\boldsymbol{\Sigma}_{T-t}^{-1} d\boldsymbol{w}_t. && (36)\\
& && (37)
\end{aligned}$$

Consequently,

$$\boldsymbol{z}_t = \boldsymbol{z}_0 + \int_0^t \sqrt{2\beta_{T-s}} e^{C_s} \boldsymbol{\Sigma}_{T-s}^{-1} d\boldsymbol{w}_s = \boldsymbol{\Sigma}_T^{-1} \boldsymbol{y}_0 + \int_0^t \sqrt{2\beta_{T-s}} e^{C_s} \boldsymbol{\Sigma}_{T-s}^{-1} d\boldsymbol{w}_s. \tag{38}$$

And,

$$\boldsymbol{y}_t = e^{-C_t} \boldsymbol{\Sigma}_{T-t} \boldsymbol{z}_t = e^{-C_t} \boldsymbol{\Sigma}_{T-t} \boldsymbol{\Sigma}_T^{-1} \boldsymbol{y}_0 + e^{-C_t} \boldsymbol{\Sigma}_{T-t} \int_0^t \boldsymbol{\Sigma}_{T-s}^{-1} e^{C_s} \sqrt{2\beta_{T-s}} d\boldsymbol{w}_s. \tag{39}$$

Finally,

$$\boldsymbol{y}_t = e^{-C_t} \boldsymbol{\Sigma}_{T-t} \boldsymbol{\Sigma}_T^{-1} \boldsymbol{y}_0 + \boldsymbol{\xi}_t \quad \text{with} \quad \boldsymbol{\xi}_t = e^{-C_t} \boldsymbol{\Sigma}_{T-t} \int_0^t \boldsymbol{\Sigma}_{T-s}^{-1} e^{C_s} \sqrt{2\beta_{T-s}} d\boldsymbol{w}_s. \tag{40}$$

By the multidimensional Itô's isometry,

$$\mathrm{Cov}\left(\int_0^t \boldsymbol{\Sigma}_{T-s}^{-1} e^{C_s} \sqrt{2\beta_{T-s}} d\boldsymbol{w}_s\right) = 2 \int_0^t e^{2C_s} \beta_{T-s} \boldsymbol{\Sigma}_{T-s}^{-2} ds. \tag{41}$$

Now, remark that for $A_s = e^{2C_s} \boldsymbol{\Sigma}_{T-s}^{-1}$,

$$\begin{aligned}
dA_s &= 2\beta_{T-s} A_s ds - e^{2C_s} d\left[\boldsymbol{\Sigma}^{-1}\right]_{T-s} && (42)\\
&= 2\beta_{T-s} A_s ds + e^{2C_s} \left[-2\beta_{T-s} \boldsymbol{\Sigma}_{T-s}^{-1} + 2\beta_{T-s} \boldsymbol{\Sigma}_{T-s}^{-2}\right] ds && (43)\\
&= 2e^{2C_s} \beta_{T-s} \boldsymbol{\Sigma}_{T-s}^{-2} ds. && (44)
\end{aligned}$$

$$\mathrm{Cov}\left(\int_0^t \boldsymbol{\Sigma}_{T-s}^{-1} e^{C_s} \sqrt{\beta_{T-s}} d\boldsymbol{w}_s\right) = \int_0^t dA_s = [A_s]_0^t = e^{2C_t} \boldsymbol{\Sigma}_{T-t}^{-1} - \boldsymbol{\Sigma}_T^{-1}. \tag{45}$$

Finally, $\mathrm{Cov}(\boldsymbol{\xi}_t) = \boldsymbol{\Sigma}_{T-t}^2 \left(\boldsymbol{\Sigma}_{T-t}^{-1} - e^{-2C_t} \boldsymbol{\Sigma}_T^{-1}\right) = \boldsymbol{\Sigma}_{T-t} - e^{-2C_t} \boldsymbol{\Sigma}_{T-t}^2 \boldsymbol{\Sigma}_T^{-1}$

We have the final formula considering:

$$C_t = \int_0^t \beta_{T-s} ds = \int_{T-t}^T \beta_x dx = \int_0^T \beta_x dx - \int_0^{T-t} \beta_x dx = B_T - B_{T-t} \tag{46}$$

that provides

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})} \boldsymbol{\Sigma}_{T-t}^2 \boldsymbol{\Sigma}_T^{-1} \left(\boldsymbol{\Sigma}_{T-t}^{-1} \mathrm{Cov}(\boldsymbol{y}_0) \boldsymbol{\Sigma}_T^{-1} \boldsymbol{\Sigma}_{T-t} - \boldsymbol{I}\right). \tag{47}$$

In particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ and $\boldsymbol{\Sigma}$ commute,

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})} \boldsymbol{\Sigma}_{T-t}^2 \boldsymbol{\Sigma}_T^{-1} \left(\boldsymbol{\Sigma}_T^{-1} \mathrm{Cov}(\boldsymbol{y}_0) - \boldsymbol{I}\right). \tag{48}$$

B.3 Proposition 3: Solution of the ODE probability flow under Gaussian assumption

As done in [16], the matrix $\boldsymbol{\Sigma}_t^{1/2}$ admits a derivative which is $d\left[\boldsymbol{\Sigma}^{1/2}\right]_t = \frac{1}{2} d\boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_t^{-1/2}$ because it is diagonalisable. Let us check that $\boldsymbol{y}_t = \boldsymbol{\Sigma}_T^{-1/2} \boldsymbol{\Sigma}_{T-t}^{1/2} \boldsymbol{y}_0$ is solution of the ODE (4),

$$\begin{aligned}
d\boldsymbol{y}_t &= -\boldsymbol{\Sigma}_T^{-1/2} \frac{1}{2} d\boldsymbol{\Sigma}_{T-t} \boldsymbol{\Sigma}_{T-t}^{-1/2} \boldsymbol{y}_0 dt && (49)\\
&= \boldsymbol{\Sigma}_T^{-1/2} \left[\beta_{T-t} \boldsymbol{\Sigma}_{T-t} - \beta_{T-t} \boldsymbol{I}\right] \boldsymbol{\Sigma}_{T-t}^{-1/2} \boldsymbol{y}_0 dt \quad \text{(using Equation(9))} && (50)\\
&= \left[\beta_{T-t} - \beta_{T-t} \boldsymbol{\Sigma}_{T-t}^{-1}\right] \boldsymbol{y}_t dt && (51)\\
&= \left[\beta_{T-t} + \beta_{T-t} \nabla_{\boldsymbol{y}} \log p_{T-t}(\boldsymbol{y}_t)\right] \boldsymbol{y}_t dt. && (52)
\end{aligned}$$

Finally,

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1/2}\boldsymbol{\Sigma}_{T-t}^{1/2}\,\mathrm{Cov}(\boldsymbol{y}_0)\boldsymbol{\Sigma}_{T-t}^{-1/2}\boldsymbol{\Sigma}_T^{1/2}. \tag{53}$$

In particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ and $\boldsymbol{\Sigma}$ commute,

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_{T-t}\,\mathrm{Cov}(\boldsymbol{y}_0). \tag{54}$$

B.4 Proof of Proposition 4

For $0 \le t \le T$, denoting $(\lambda_i^t)_{1\le i\le d}$ the eigenvalues of $\boldsymbol{\Sigma}_t$, the eigenvalues of $\tilde{\boldsymbol{\Sigma}}_t = \mathrm{Cov}(\tilde{\boldsymbol{y}}_{T-t})$ are

$$\tilde{\lambda}_i^t = \lambda_i^t + e^{-2(B_T-B_t)}\left(\lambda_i^t\right)^2\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right), \quad i=1,\dots,d. \tag{55}$$

and the eigenvalues of $\widehat{\boldsymbol{\Sigma}}_t = \mathrm{Cov}(\widehat{\boldsymbol{y}}_{T-t})$ are

$$\widehat{\lambda}_i^t = \frac{\lambda_i^t}{\lambda_i^T}, \quad i=1,\dots,d. \tag{56}$$

Consequently, $\mathbf{W}_2(p_t,\tilde{p}_t)$ is the sum of the squares of all:

$$\sqrt{\lambda_i^t}-\sqrt{\tilde{\lambda}_i^t} = \sqrt{\lambda_i^t}\left(1-\sqrt{1+e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right)}\right). \tag{57}$$

Similarly, $\mathbf{W}_2(p_t,\widehat{p}_t)$ is the sum of the squares of all:

$$\sqrt{\lambda_i^t}-\sqrt{\widehat{\lambda}_i^t} = \sqrt{\lambda_i^t}\left(1-\sqrt{\frac{1}{\lambda_i^T}}\right) \tag{58}$$

$$= \sqrt{\lambda_i^t}\left(1-\sqrt{1+\left(\frac{1}{\lambda_i^T}-1\right)}\right). \tag{59}$$

Let us now compare individually these differences.

$$\frac{e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right)}{\frac{1}{\lambda_i^T}-1} = e^{-2(B_T-B_t)}\frac{\lambda_i^t}{\lambda_i^T} \tag{60}$$

$$= e^{-2(B_T-B_t)}\frac{e^{-2B_t}(\lambda_i-1)+1}{e^{-2B_T}(\lambda_i-1)+1} \tag{61}$$

$$= \frac{(\lambda_i-1)+e^{2B_t}}{(\lambda_i-1)+e^{2B_T}} \tag{62}$$

$$< 1. \tag{63}$$

Case 1: $0<\lambda_i<1$ **and** $t>0$

In this case, $\lambda_i^T<1$ and:

$$0 < e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right) < \frac{1}{\lambda_i^T}-1. \tag{64}$$

Thus,

$$\left|\sqrt{\lambda_i^t}-\sqrt{\tilde{\lambda}_i^t}\right| = \sqrt{\tilde{\lambda}_i^t}-\sqrt{\lambda_i^t} \tag{65}$$

$$= \sqrt{\lambda_i^t}\left(\sqrt{1+e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right)}-1\right) \tag{66}$$

$$< \sqrt{\lambda_i^t}\left(\sqrt{1+\left(\frac{1}{\lambda_i^T}-1\right)}-1\right) \tag{67}$$

$$= \sqrt{\widehat{\lambda}_i^t}-\sqrt{\lambda_i^t} \tag{68}$$

$$= \left|\sqrt{\lambda_i^t}-\sqrt{\widehat{\lambda}_i^t}\right|. \tag{69}$$

Case 2: $\lambda_i = 0$ **and** $t = 0$**.**

In this case, for $1 \leq i \leq d$, $\widehat{\lambda}_i^T = \tilde{\lambda}_i^T = 0$.

Case 3: $\lambda_i = 1$**.**

In this case, for $1 \leq i \leq d$, $\widehat{\lambda}_i^t = \tilde{\lambda}_i^t = 1$.

Case 4: $1 < \lambda_i$**.**

In this case, $\lambda_i^T \geq 1$, and $\frac{e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right)}{\frac{1}{\lambda_i^T}-1} = e^{-2(B_T-B_t)}\frac{\lambda_i^t}{\lambda_i^T} < 1$ provides

$$e^{-2(B_T-B_t)}\lambda_i^t\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right) > \frac{1}{\lambda_i^T}-1. \tag{70}$$

Finally,

$$\left|\sqrt{\lambda_i^t}-\sqrt{\tilde{\lambda}_i^t}\right| = \sqrt{\lambda_i^t}-\sqrt{\tilde{\lambda}_i^t} \tag{71}$$

$$= \sqrt{\lambda_i^t}\left(1-\sqrt{1+e^{-2(B_T-B_t)}\lambda_i^T\frac{1}{\lambda_i^T}\left(\frac{1}{\lambda_i^T}-1\right)}\right) \tag{72}$$

$$< \sqrt{\lambda_i^t}\left(1-\sqrt{1+\left(\frac{1}{\lambda_i^T}-1\right)}\right) \tag{73}$$

$$= \sqrt{\lambda_i^t}-\sqrt{\widehat{\lambda}_i^t} \tag{74}$$

$$= \left|\sqrt{\lambda_i^t}-\sqrt{\widehat{\lambda}_i^t}\right|. \tag{75}$$

This case study provides:

$$\mathbf{W}_2(\tilde{p}_t, p_t) \leq \mathbf{W}_2(\widehat{p}_t, p_t). \tag{76}$$

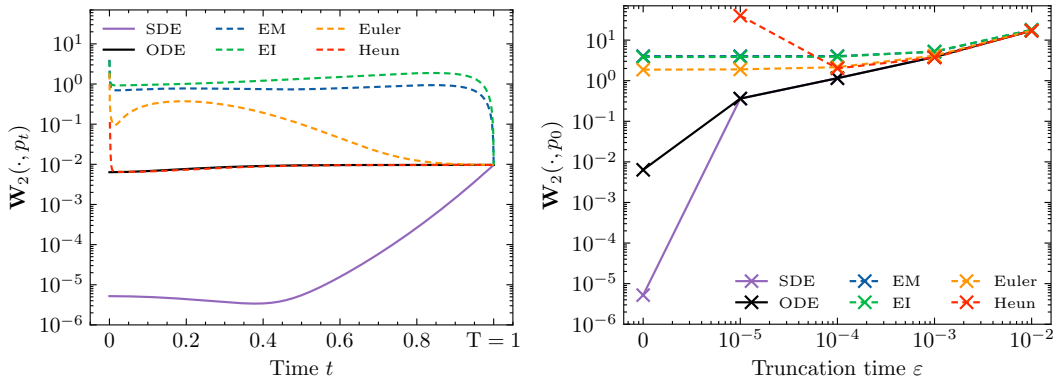
C Gaussian CIFAR-10 samples

The Gaussian CIFAR-10 produces unstructured images. A grid of samples is presented in Figure 4.

Figure 4: **CIFAR-10 Gaussian samples**. Samples are generated from the Gaussian distribution fitting the CIFAR-10 dataset.

D Theoretical Wasserstein distance for the ADSN model

As done for the Gaussian CIFAR-10, the Wasserstein errors can be computed for the ADSN model as shown in Figure 5 and Table 4.



(a) Initialization error along the integration time.

(b) Truncation error for different truncation time ε.

Figure 5: **Wasserstein errors for the diffusion models associated with the Gaussian microtextures.** Left: Evolution of the Wasserstein distance between p_t and the distributions associated with the continuous SDE, the continuous flow ODE and four discrete sampling schemes with standard $\mathcal{N}_0$ initialization, either stochastic (Euler-Maruyama (EM) and Exponential Integrator (EI)) or deterministic (Euler and Heun). While the continuous SDE is less sensible than the continuous ODE (as proved by Proposition 4), the initialization error impacts all discrete schemes. Heun's method has the lowest error and is very close to the theoretical ODE, except for the last step that is usually discarded when using time truncation. Right: Wasserstein errors due to time truncation for various truncation times ε. Heun's scheme is not defined without truncation time due to the zero eigenvalue. Interestingly, for the standard practice truncation time $\varepsilon = 10^{-3}$, all numerical schemes have a comparable error close to their continuous counterparts.

		Continuous		$N = 50$		$N = 250$		$N = 500$		$N = 1000$	
		p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$
EM	$\varepsilon = 0$	0	5.2E-6	53.37	53.37	10.58	10.58	6.27	6.27	4.02	4.02
	$\varepsilon = 10^{-5}$	0.36	0.36	53.35	53.35	10.57	10.57	6.26	6.26	4.02	4.02
	$\varepsilon = 10^{-3}$	3.84	3.84	51.92	51.92	10.55	10.55	6.80	6.80	5.16	5.16
	$\varepsilon = 10^{-2}$	17.09	17.09	48.24	48.24	20.39	20.39	18.57	18.57	17.79	17.79
EI	$\varepsilon = 0$	0	5.2E-6	30.91	30.91	8.85	8.85	5.71	5.71	3.84	3.84
	$\varepsilon = 10^{-5}$	0.36	0.36	30.92	30.92	8.85	8.85	5.72	5.72	3.84	3.84
	$\varepsilon = 10^{-3}$	3.84	3.84	31.94	31.94	9.74	9.74	6.76	6.76	5.24	5.24
	$\varepsilon = 10^{-2}$	17.09	17.09	41.49	41.49	21.02	21.02	18.95	18.95	17.99	17.99
Euler	$\varepsilon = 0$	0	6.4E-3	5.69	5.70	3.27	3.27	2.50	2.51	1.87	1.87
	$\varepsilon = 10^{-5}$	0.36	0.36	5.70	5.71	3.28	3.28	2.53	2.53	1.90	1.90
	$\varepsilon = 10^{-3}$	3.84	3.84	6.79	6.79	4.85	4.85	4.41	4.41	4.14	4.14
	$\varepsilon = 10^{-2}$	17.09	17.09	18.52	18.52	17.35	17.35	17.22	17.22	17.15	17.15
Heun	$\varepsilon = 0$	0	6.4E-3	-	-	-	-	-	-	-	-
	$\varepsilon = 10^{-5}$	0.36	0.36	2.4E+3	2.4E+3	3.0E+2	3.0E+2	1.1E+2	1.1E+2	40.00	40.00
	$\varepsilon = 10^{-3}$	3.84	3.84	15.42	15.42	2.25	2.25	3.40	3.40	3.73	3.73
	$\varepsilon = 10^{-2}$	17.09	17.09	16.59	16.59	17.07	17.07	17.09	17.09	17.09	17.09

Table 4: **Ablation study of Wasserstein errors for the Gaussian microtextures.** For a given discretization scheme, the table presents the Wasserstein distance associated with the truncation error for different values of ε. The columns p_T and $\mathcal{N}_0$ show the influence of the initialization error. The continuous column corresponds to the continuous SDE or ODE linked with the scheme (identical values for EM, EI and Euler, Heun). Note that the Heun scheme is not defined without truncation time due to the zero eigenvalue.

E Study of the covariance matrix of the ADSN distribution

E.1 Reminders on the Discrete Fourier Transform (DFT)

For a given image $\boldsymbol{v} \in \mathbb{R}^{3\times M\times N}$, we define the DFT of $\boldsymbol{v}$, $\widehat{\boldsymbol{v}} \in \mathbb{R}^{3\times M\times N}$ such that for $1 \leq c \leq 3$,$\xi \in \mathbb{R}^{M\times N}$

$$\widehat{\boldsymbol{v}}_{c,\xi} = \sum_{x\in M\times N} \boldsymbol{v}_{c,x} \exp(-\frac{2i\pi x_1\xi_1}{M}) \exp(-\frac{2i\pi x_2\xi_2}{N}), \quad i^2 = -1 \tag{77}$$

where $\widehat{\boldsymbol{v}}_{c,\xi}$ is the value of $\widehat{\boldsymbol{v}}$ at coordinate ξ of the k-th channel of $\widehat{\boldsymbol{v}}$. For $\boldsymbol{u} \in \mathbb{R}^{3M\times N}$, by defining $\boldsymbol{u} \star \boldsymbol{v}$ the periodic convolution such that for $1 \leq c \leq 3, x \in \mathbb{R}^{M\times N}$:

$$(\boldsymbol{u} \star \boldsymbol{v})_{c,x} = \sum_{y\in M\times N} \boldsymbol{u}_{c,x-y}\boldsymbol{v}_{c,y} \tag{78}$$

we have:

$$\widehat{\boldsymbol{u} \star \boldsymbol{v}} = \widehat{\boldsymbol{u}} \odot \widehat{\boldsymbol{v}}, \tag{79}$$

where $\odot$ is the componentwise product.

E.2 Eigenvectors of the covariance matrix of the ADSN distribution

Let $\boldsymbol{u} \in \mathbb{R}^{3\times M\times N}$ and its associated texton $\mathbf{t} \in \mathbb{R}^{3\times M\times N}$. The distribution $\mathrm{ADSN}(\boldsymbol{u})$ is the Gaussian distribution of $\boldsymbol{X} = \mathbf{t} \star \boldsymbol{w}$ such that:

$$\boldsymbol{X}_i = \mathbf{t}_i \star \boldsymbol{w} \in \mathbb{R}^{M\times N}, 1 \leq i \leq 3, \boldsymbol{w} \sim \mathcal{N}_0 \tag{80}$$

Consequently, denoting $\boldsymbol{\Sigma}$ the covariance of $\mathrm{ADSN}(\boldsymbol{u})$, for $\boldsymbol{v} \in \mathbb{R}^{3M\times N}$,

$$\widehat{\boldsymbol{\Sigma}\boldsymbol{v}}_i = \widehat{\mathbf{t}}_i\overline{\widehat{\mathbf{t}}_1}\widehat{\boldsymbol{v}}_1 + \widehat{\mathbf{t}}_i\overline{\widehat{\mathbf{t}}_2}\widehat{\boldsymbol{v}}_2 + \widehat{\mathbf{t}}_i\overline{\widehat{\mathbf{t}}_3}\widehat{\boldsymbol{v}}_3 = \widehat{\mathbf{t}}_i \left(\overline{\widehat{\mathbf{t}}_1}\widehat{\boldsymbol{v}}_1 + \overline{\widehat{\mathbf{t}}_2}\widehat{\boldsymbol{v}}_2 + \overline{\widehat{\mathbf{t}}_3}\widehat{\boldsymbol{v}}_3\right) \tag{81}$$

This equation proves that the kernel of $\boldsymbol{\Sigma}$ contains the kernel of $\boldsymbol{v} \in \mathbb{R}^{3\times M\times N} \mapsto \overline{\widehat{\mathbf{t}}_1}\widehat{\boldsymbol{v}}_1 + \overline{\widehat{\mathbf{t}}_2}\widehat{\boldsymbol{v}}_2 + \overline{\widehat{\mathbf{t}}_3}\widehat{\boldsymbol{v}}_3 \in \mathbb{R}^{M\times N}$ which has a dimension greater than $2MN$. Consequently, 0 is eigenvalue of $\boldsymbol{\Sigma}$ with multiplicity greater than $2MN$. Furthermore, for $\xi \in \mathbb{R}^{M\times N}$, denoting $\boldsymbol{u}^{1,\xi}$ such that:

$$\widehat{\boldsymbol{u}}_i^{1,\xi}(\omega) = \mathbf{1}_{\omega=\xi}\widehat{\mathbf{t}}_i(\omega), 1 \leq i \leq 3, \omega \in \mathbb{R}^{M\times N} \tag{82}$$

we have,

$$\boldsymbol{\Sigma}\boldsymbol{u}^{1,\xi} = (|\widehat{\mathbf{t}}_1(\xi)|^2 + |\widehat{\mathbf{t}}_2(\xi)|^2 + |\widehat{\mathbf{t}}_3(\xi)|^2)\boldsymbol{u}^{1,\xi}. \tag{83}$$

Furthermore, the family $\left(\boldsymbol{u}^{1,\xi}\right)_{\xi\in M\times N}$ is orthogonal. Thus, the eigenvalues of $\boldsymbol{\Sigma}$ are $\left(|\widehat{\mathbf{t}}_1(\xi)|^2 + |\widehat{\mathbf{t}}_2(\xi)|^2 + |\widehat{\mathbf{t}}_3(\xi)|^2\right)_{\xi\in M\times N}$ and 0 with multiplicity $2MN$.

For $\xi \in \mathbb{R}^{M\times N}$, we denote $\boldsymbol{u}^{2,\xi}$,$\boldsymbol{u}^{3,\xi}$ such that for $\omega \in \mathbb{R}^{M\times N}$:

$$\begin{cases} \widehat{\boldsymbol{u}}_1^{2,\xi}(\omega) &= -\mathbf{1}_{\omega=\xi}\overline{\widehat{\mathbf{t}}_3}(\omega) \\ \widehat{\boldsymbol{u}}_2^{2,\xi}(\omega) &= 0 \\ \widehat{\boldsymbol{u}}_3^{2,\xi}(\omega) &= \mathbf{1}_{\omega=\xi}\overline{\widehat{\mathbf{t}}_1}(\omega) \end{cases} \tag{84}$$

$$\begin{cases} \widehat{\boldsymbol{u}}_1^{3,\xi}(\omega) &= 0 \\ \widehat{\boldsymbol{u}}_2^{3,\xi}(\omega) &= -\mathbf{1}_{\omega=\xi}\overline{\widehat{\mathbf{t}}_3}(\omega) \\ \widehat{\boldsymbol{u}}_3^{3,\xi}(\omega) &= \mathbf{1}_{\omega=\xi}\overline{\widehat{\mathbf{t}}_2}(\omega) \end{cases} \tag{85}$$

We have

$$\boldsymbol{\Sigma}\boldsymbol{u}^{2,\xi} = 0.\boldsymbol{u}^{2,\xi} \tag{86}$$

$$\boldsymbol{\Sigma}\boldsymbol{u}^{3,\xi} = 0.\boldsymbol{u}^{3,\xi}. \tag{87}$$

Then, applying the orthonomalization of Gram-Schmidt on each tuple $(\boldsymbol{u}^{1,\xi}, \boldsymbol{u}^{2,\xi}, \boldsymbol{u}^{3,\xi})_{\xi\in\mathbb{R}^{M\times N}}$, we obtain an orthonormal basis in the Fourier domain $(\boldsymbol{v}^{1,\xi}, \boldsymbol{v}^{2,\xi}, \boldsymbol{v}^{3,\xi})_{\xi\in\mathbb{R}^{M\times N}}$ of eigenvectors of $\boldsymbol{\Sigma}$. More precisely, for $\xi_1, \xi_2 \in \mathbb{R}^{M\times N}$, $1 \leq j_1, j_2 \leq 3$,

$$\left(\overline{\widehat{\boldsymbol{v}}}^{j_1,\xi_1}\right)^T \widehat{\boldsymbol{v}}^{j_2,\xi_2} = \sum_{\substack{x_1\in M\times N\\ x_2\in M\times N}} \overline{\widehat{\boldsymbol{v}}}^{j_1,\xi_1}_{x_1} \widehat{\boldsymbol{v}}^{j_2,\xi_2}_{x_2} \tag{88}$$

$$= \mathbf{1}_{\substack{j_1=j_2\\ \xi_1=\xi_2}} \tag{89}$$

which is applying the square root of $\boldsymbol{\Sigma}$ to the white Gaussian noise $\boldsymbol{w}$. Furthermore, we can ensure that for $\xi \neq \omega \in \mathbb{R}^{M\times N}, 1 \leq j \leq 3$, $\widehat{\boldsymbol{v}}^{j,\xi}(\omega) = 0$ such that only the frequency ξ is active in the Fourier transform of $\boldsymbol{v}^{j,\xi}$. Consequently, for $\boldsymbol{w} \in \mathbb{R}^{3M\times N}$,

$$\overline{\widehat{\boldsymbol{w}}}^T \boldsymbol{v}^{j,\xi} = \sum_{1\leq i\leq 3} \overline{\widehat{\boldsymbol{w}}}_i(\xi)\widehat{\boldsymbol{v}}_i^{j,\xi}(\xi). \tag{90}$$

In particular,

$$\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T = \|\widehat{\boldsymbol{v}}^{j,\xi}\|^2 = \sum_{1\leq i\leq 3} \left|\boldsymbol{v}_i^{j,\xi}(\xi)\right|^2 = 1. \tag{91}$$

E.3 Computation of the empirical Wasserstein error in the ADSN covariance diagonalization basis

Let consider a Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ such that there exists $(\lambda_1^\xi, \lambda_2^\xi, \lambda_3^\xi)_{\xi\in\mathbb{R}^{M\times N}}$ such that for all $\xi \in \mathbb{R}^{M\times N}$,

$$\boldsymbol{\Gamma}\boldsymbol{v}^{j,\xi} = \lambda_j^\xi \boldsymbol{v}^{j,\xi}, \quad 1 \leq j \leq 3. \tag{92}$$

Let $\boldsymbol{w} \sim \mathcal{N}_0 \in \mathbb{R}^{3M\times N}$, $(\boldsymbol{v}^{1,\xi}, \boldsymbol{v}^{2,\xi}, \boldsymbol{v}^{3,\xi})_{\xi\in\mathbb{R}^{M\times N}}$ is an orthonormal basis in the Fourier domain such that:

$$\widehat{\boldsymbol{w}} = \sum_{\xi\in\mathbb{R}^{M\times N}} \left(\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{1,\xi}\right]\widehat{\boldsymbol{v}}^{1,\xi} + \left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{2,\xi}\right]\widehat{\boldsymbol{v}}^{2,\xi} + \left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{3,\xi}\right]\widehat{\boldsymbol{v}}^{3,\xi}\right) \tag{93}$$

$$\tag{94}$$

A sample drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ has the same distribution as $\boldsymbol{Y}$ given by

$$\widehat{\boldsymbol{Y}} = \sum_{\xi\in\mathbb{R}^{M\times N}} \sqrt{\lambda_1^\xi}\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{1,\xi}\right]\widehat{\boldsymbol{v}}^{1,\xi} + \sum_{\xi\in\mathbb{R}^{M\times N}} \sqrt{\lambda_2^\xi}\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{2,\xi}\right]\widehat{\boldsymbol{v}}^{2,\xi} + \sum_{\xi\in\mathbb{R}^{M\times N}} \sqrt{\lambda_3^\xi}\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{3,\xi}\right]\widehat{\boldsymbol{v}}^{3,\xi}. \tag{95}$$

Note that the three channels of $\boldsymbol{w}$ are independent. Furthermore, for $1 \leq j \leq 3$

$$\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T \widehat{\boldsymbol{Y}} = \sqrt{\lambda_1^\xi}\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{j,\xi}\right]\left\|\widehat{\boldsymbol{v}}^{j,\xi}\right\|^2 = \sqrt{\lambda_1^\xi}\left[\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{j,\xi}\right] \tag{96}$$

$$\left|\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T \widehat{\boldsymbol{Y}}\right|^2 = \lambda_j^\xi \left|\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{j,\xi}\right|^2 \tag{97}$$

$$\mathbb{E}\left[\left|\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T \widehat{\boldsymbol{Y}}\right|^2\right] = \lambda_j^\xi \mathbb{E}\left[\left|\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{j,\xi}\right|^2\right] \tag{98}$$

$$\mathbb{E}\left[\left|\overline{\widehat{\boldsymbol{w}}}^T \widehat{\boldsymbol{v}}^{j,\xi}\right|^2\right] = \sum_{1\leq c_1,c_2\leq 3} \mathbb{E}\left[\overline{\widehat{\boldsymbol{w}}}_{c_1}(\xi)\widehat{\boldsymbol{w}}_{c_2}(\xi)\right]\widehat{\boldsymbol{v}}_{c_1}^{j,\xi}(\xi)\overline{\widehat{\boldsymbol{v}}}_{c_2}(\xi) \text{ by Equation (90)} \tag{99}$$

$$= \sum_{1\leq c\leq 3} \mathbb{E}\left[\left|\widehat{\boldsymbol{w}}_c(\xi)\right|^2\right]\left|\widehat{\boldsymbol{v}}_c^{j,\xi}(\xi)\right|^2 \text{ because the channels are inependent} \tag{100}$$

$$= 3MN \sum_{1\leq c\leq 3} \left|\widehat{\boldsymbol{v}}_c^{j,\xi}(\xi)\right|^2 \text{ because } \mathbb{E}\left[\left|\widehat{\boldsymbol{w}}_c(\xi)\right|^2\right] = MN \tag{101}$$

$$= 3MN \text{ by Equation (91).} \tag{102}$$

Finally,

$$\mathbb{E}\left[\left|\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T \widehat{\boldsymbol{Y}}\right|^2\right] = 3MN\lambda_1^\xi \tag{103}$$

Finally, for a given sampling $(\boldsymbol{Y}_k)_{1\leq k\leq N_{\text{samples}}}$ following the distribution $\mathcal{N}(\mathbf{0},\boldsymbol{\Gamma})$, an estimator of λ_j^ξ is:

$$\lambda_j^{\xi,\text{emp.}} = \frac{1}{3N_{\text{samples}}MN}\sum_{k=1}^{N_{\text{samples}}}\left|\left(\overline{\widehat{\boldsymbol{v}}}^{j,\xi}\right)^T \widehat{\boldsymbol{Y}_k}\right|^2. \tag{104}$$

The empirical Wasserstein distance between the Gaussian distribution $\mathcal{N}(\mathbf{0},\Gamma)$ and the ADSN model with texton $\mathbf{t}$ is:

$$\mathbf{W}_2^{\text{emp.}}(\mathcal{N}^{\text{emp.}}(\mathbf{0},\boldsymbol{\Gamma}),\text{ADSN}(\boldsymbol{u})) = \sqrt{\sum_{\xi\in\mathbb{R}^{M\times N}}\left(\left(\sqrt{\lambda_1^{\xi,\text{emp.}}}-\sqrt{\lambda_1^{\xi,\text{ADSN}}}\right)^2+\lambda_2^{\xi,\text{emp.}}+\lambda_3^{\xi,\text{emp.}}\right)} \tag{105}$$

with $\lambda_1^{\xi,\text{ADSN}} = |\widehat{\mathbf{t}}_1(\xi)|^2 + |\widehat{\mathbf{t}}_2(\xi)|^2 + |\widehat{\mathbf{t}}_3(\xi)|^2$ for $\xi\in\mathbb{R}^{M\times N}$.

Furthermore, the computations can be vectorized by componentwise products in the Fourier domain.