Generative Diffusion Models for High-Dimensional Time Series

Riya Danait
Mathematical Institute
University of Oxford
riya.danait@maths.ox.ac.uk

Rama Cont
Mathematical Institute
University of Oxford
rama.cont@maths.ox.ac.uk

Abstract

We provide a two-stage approach for high dimensional time series generation: (i) kernel estimation for the conditional first and second moments of the underlying data increments to recover residuals, and (ii) score-based diffusion trained on these residuals. We give finite-time convergence estimates for the reverse SDE in total variation (TV) and Wasserstein-2 (W_2), with explicit dependence on the variance preserving noise schedule, a corrected initial mismatch of Gaussian targets, and a Grönwall coupling that separates initialization, score and discretization errors. Experiments on synthetic multivariate processes validate: (a) empirical TV and W_2 track the theoretical upper bounds, and (b) Monte Carlo estimates of test functionals achieve the predicted standard errors.

1 Introduction

Time-reversed diffusion models have emerged as an interesting approach to generative modeling (Sohl-Dickstein et al. [2015], Song and Ermon [2019], Ho et al. [2020], Song et al. [2021]), achieving significant empirical success in image, audio, and text synthesis, of which DALL-E and SORA are perhaps the most well-known examples. There are two main types of diffusion models: denoising diffusion probabilistic models (DDPMs) (Ho et al. [2020], Dhariwal and Nichol [2021]) and denoising diffusion implicit models (DDIMs) Song et al. [2020], in which the diffusion processes are non-Markovian. We utilize DDPMs to motivate our methodology.

DDPMs are comprised of a forward process and a reverse process. The forward *noising* process is characterized by a stochastic differential equation (SDE) initialized using the empirical distribution of a data sample. The forward distribution is often chosen to be ergodic, with a known stationary distribution. Given the forward process, we can construct a corresponding time-reversed process, called the *denoising* process. To generate samples from the target data distribution, we simulate the reverse process starting from an I.I.D. initialization with a Gaussian distribution.

Generative modeling for multivariate time series poses multiple challenges, particularly preserving complex temporal structure. It is not enough to learn the marginal distribution or even the joint distribution without exploiting the sequential nature of the data. We require a conditional generative model that generates each observation considering the past observations. Recent time-series generators have introduced more powerful techniques involving Generative Adversarial Networks (GANs) (Yoon et al. [2019], Vuletić et al. [2024]) and Variational Autoencoders (VAEs) Bühler et al. [2020]. Diffusion models have also driven much of the progress for time series tasks such as imputation and forecasting (Rasul et al. [2021], Kollovieh et al. [2023], Yang et al. [2024], Yuan and Qiao [2024], Su et al. [2025]).

We introduce an algorithm that involves a Nadaraya-Watson kernel estimator to decompose the time series into its conditional mean, covariance and residuals, followed by training a score-based diffusion

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Generative AI in Finance.

model on these extracted residuals. Our convergence analysis is complementary to recent work on (i) generalization of learned scores Stéphanovitch et al. [2025], (ii) regularity beyond log-concavity (Stéphanovitch [2025], Gentiloni-Silveri and Ocello [2025]), and (iii) explicit KL/W_2 for score-based generative model families Conforti et al. [2024] and noise-schedule sensitivity analysis Strasman et al. [2025]. The TV and W_2 bounds that we provide are novel in that they make the dependence on the noise schedule explicit and decouple initialization, score, and discretization errors via a Grönwall coupling.

2 Description of Algorithm

Let $X_{t_k} \in \mathbb{R}^d$ denote the observations, where $t_k = k\Delta t,\ k = 1, \ldots, N$ with Δt timesteps. We utilize the Nadaraya-Watson kernel estimator Nadaraya [1964], Watson [1964], Nadaraya [1970] to approximate the conditional mean and covariance structure of our data samples:

$$\begin{aligned} \mu(x) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{E}[\Delta X_t \mid X_t = x] \\ a(x) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathrm{Cov}(\Delta X_t \mid X_t = x), \end{aligned} \tag{1}$$

where $a(x) = \sigma^\top \sigma(x) \in \mathbb{R}^{d \times d}$ is the conditional covariance matrix of the increments. The estimators are given by:

$$\widehat{\mu}(x) = \frac{\sum_{k=1}^N K_h(x - X_{t_k}) \Delta X_{t_k}}{W(x)} \tag{2}$$

$$\widehat{a}(x) = \frac{\sum_{k=1}^N K_h(x - X_{t_k})(\Delta X_{t_k} - \hat{\mu}(x))(\Delta X_{t_k} - \hat{\mu}(x))^\top}{W(x)}, \tag{3}$$

where $W(x) = \Delta t \sum_{k=1}^N K_h(x - X_{t_k})$ for $K_h(x)$ kernel function with bandwidth h, and $\Delta X_{t_k} = X_{t_{k+1}} - X_{t_k}$. The bandwidth h is chosen in a locally adaptive k nearest neighbors manner. Define now $\widehat{\sigma}(x)$ as a Cholesky square root of $\widehat{a}(x)$:

$$\widehat{\sigma}^\top(x)\widehat{\sigma}(x) = \widehat{a}(x) \tag{4}$$

We may define the *residuals*

$$\widehat{\epsilon}_{t_i}^{(n)} = \widehat{\sigma}^\top (X_{t_i}^{(n)})^{-1} [\Delta X_{t_i}^{(n)} - \widehat{\mu}(X_{t_i}^{(n)})]. \tag{5}$$

Remark 1. *Note that, as the square root of the matrix $\widehat{a}$ is only defined up to a rotation, we cannot hope to recover a consistent estimator of $\sigma(x)$ i.e that $\widehat{\sigma}(x) \to \sigma(x)$. However, as we will see, under high-frequency asymptotics on the observed path we will typically have $\widehat{a}(x) \to a(x)$ i.e. we recover $\sigma(x)$ up to a (local) rotation. This means we cannot interpret the ϵ_{t_k} as a "filtering" of the noise terms, but these residuals allow us to recover, asymptotically, the second order structure of ϵ_t.*

For the score-based diffusion model, we use a time dependent Ornstein-Uhlenbeck (OU) process for the forward SDE:

$$\begin{array}{rcl} \mathrm{d}X_t & = & -\frac{1}{2}\beta_t X_t \mathrm{d}t + \sqrt{\beta_t} dW_t \\ X_0 & \sim & p_0, \end{array} \tag{6}$$

where β_t is a time-dependent function. Let us define $\alpha_t = \int_0^t \beta_s \mathrm{d}s$. Then the reverse SDE is given by

$$\begin{array}{rcl} \mathrm{d}Y_t & = & \frac{1}{2}\beta_{T-t} Y_t \mathrm{d}t + \beta_{T-t} \nabla \log p_{T-t}(Y_t) \mathrm{d}t + \sqrt{\beta_{T-t}} \mathrm{d}W_t, \\ Y_0 & \sim & \mathcal{N}(m_T x_0, v_T I), \end{array} \tag{7}$$

where $m_t = \exp(-\frac{1}{2}\alpha_t)$ and $v_t = 1 - \exp(-\alpha_t)$. Note that $X_t \stackrel{d}{=} m_t X_0 + \sqrt{v_t}\epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$, so that the *exact* score function is

$$\nabla \log p_{t|0}(x \mid x_0) = \frac{m_t x_0 - x}{v_t} \stackrel{d}{=} -\frac{\epsilon}{\sqrt{v_t}}. \tag{8}$$

We use 8 as the conditional target for training our score network. See Appendix A for more details. Once we have determined the estimators, we can filter the noise and feed it into the score-based diffusion model to generate new samples. Algorithm 1 shows the steps of the algorithm.

Algorithm 1 Training of generative model for time series

Input: Observations $X_{t_k} \in \mathbb{R}^d$ with $k = 1, ..., N$ where $t_k = k\Delta t$, $\Delta X_{t_k} = X_{t_{k+1}} - X_{t_k}$.
for $x \in \mathbb{D} \subset \mathbb{R}^d$ **do** ▷ Kernel Estimation
Compute weight denominator $W(x) = \Delta t \sum_{k=1}^{N} K_h(x - X_{t_k})$ for $K_h(x)$ kernel function with bandwidth h.
Compute $\widehat{\mu}(x) = \frac{\sum_{k=1}^{N} K_h(x - X_{t_k})\Delta X_{t_k}}{W(x)}$.
Compute $\widehat{a}(x) = \frac{\sum_{k=1}^{N} K_h(x - X_{t_k})(\Delta X_{t_k} - \hat{\mu}(x))(\Delta X_{t_k} - \hat{\mu}(x))^\top}{W(x)}$, $\widehat{\sigma}(x) = \texttt{CholeskySqrt}(\widehat{a}(x))$.
end for
for $k = 1$ to N **do** ▷ Residuals
$\epsilon_{t_k} = \widehat{\sigma}^\top(X_{t_k})^{-1}[\Delta X_{t_k} - \widehat{\mu}(X_{t_k})]$
end for
▷ Offline: learning to generate the residuals
Precompute **noise schedule** $\beta_t = \beta_{\text{max}}^{1-t}\beta_{\text{min}}^{t}$, $m_t = \exp(-0.5\int_0^t \beta_s \mathrm{d}s)$, and $v_t = 1 - m(t)^2$.
while current_iteration < Max_iterations **do**
Sample a minibatch $\{x_0^{(b)}, b \in B\} \subset \{\widehat{\epsilon_{t_k}}, k = 1, \ldots, N\}$, $(t^{(b)} \sim \texttt{UNIF}[0, 1], b \in B)$.
For $b \in B$, set $x_t^{(b)} = m_{t^{(b)}} x_0^{(b)} + \sqrt{v_{t^{(b)}}} z^{(b)}$ where $(z^{(b)} \sim \mathcal{N}(0, I), b \in B)$ are IID.
Compute "score targets" $u_{t^{(b)}} = -z^{(b)}/\sqrt{v_{t^{(b)}}} = \nabla \log p_{t|0}(x_t^{(b)} \mid x_0^{(b)})$.
Compute batch loss function

$$\mathcal{L}_B(\theta) = \frac{1}{|B|} \sum_{b \in B} \| s_\theta(x_t^{(b)}, t^{(b)}) - u_t^{(b)} \|^2.$$

Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$.
end while
Outputs: $\widehat{\mu}, \widehat{\sigma}$ and trained score function s_θ^*
To generate a sample path $(\widehat{X}(t_k), k = 1, \ldots, N)$: ▷ Generation of sample paths
for $j = 1, \ldots, N$ **do**
Simulate discretized paths for the (reverse) SDE on the grid $(u_i = i/m, i = 0, \ldots, m)$.
$Y_0 \sim N(m_T X_0, v_T I)$
for $i = 0, \ldots, m$ **do**

$$Y_{u_{i+1}} = Y_{u_i} + \frac{1}{m}\left(\frac{1}{2}\beta_{T-u_i} Y_{u_i} + \beta_{T-u_i} s_\theta^*(Y_{u_i}, T - u_i)\right) + \sqrt{\beta_{T-u_i}/m}\, Z_i, \qquad Z_i \overset{iid}{\sim} N(0, I)$$

end for
$\widehat{\epsilon}_j \leftarrow Y_T$
end for

for $j = 1, \ldots N - 1$ **do**

$$\widehat{X_{t_{j+1}}} = \widehat{X_{t_j}} + \widehat{\mu}(\widehat{X_{t_j}})(t_{j+1} - t_j) + \widehat{\sigma}(\widehat{X_{t_j}})\widehat{\epsilon}_j$$

end for
return Synthetic samples $\{\widehat{X_{t_k}}, k = 1, \ldots, N\}$

Remark 2. *Our nonparametric estimation captures temporal dependence to the extent it is included in the conditioning set. With state-only inputs, the model is effectively Markov; for non-Markov data, the kernel inputs can be augmented with lagged covariates and a past-only adaptive k-NN bandwidth.*

3 Convergence Analysis

When examining the convergence of the reverse process, we start first by making the following assumption regarding score matching:

Assumption 1. *For some $0 \leq t \leq T$, $\epsilon_{score} > 0$, we have access to score estimates $s_\theta(\cdot)$ satisfying $\mathbb{E}_{p_t}[\|s_\theta(X_t, t) - \nabla \log p_t(X_t, t)\|^2] \leq \epsilon^2_{score}$.*

De Bortoli et al. [2021] provided a first bound for $TV(\text{Law}(Y_T), p_0(\cdot))$, with the work of Chen et al. [2023] improving the bound to be polynomial in dimension d and time T. Under the above assumption 1, if we apply the total variation distance to our setting, we obtain

$$TV(\text{Law}(Y_T), p_0(\cdot)) \leq m_T \left(\sqrt{\mathbb{E}_{p_0}[|X_0|^2]}/2 \right) + \epsilon_{\text{score}} \sqrt{T/2}. \tag{9}$$

We expand our convergence results by including Wasserstein bounds. First, we can make a stronger assumption on the score matching, i.e.

Assumption 2. *For some $0 \leq t \leq T$, $\epsilon_{score} > 0$, we have access to score estimates $s_\theta(\cdot)$ satisfying $\mathbb{E}_{p_t}[\|s_\theta(X_t, t) - \nabla \log p_t(X_t, t)\|_\infty] \leq \epsilon_{score}$.*

We require an assumption on the growth of the drift coefficient and regularity of the score function:

Assumption 3. *Recall the forward SDE* (11)*. Then*

- $\exists\ \rho(t) : [0, T] \to \mathbb{R}$ *such that* $(x - y)(f(x, t) - f(y, t)) \geq \rho(t)|x - y|^2$.
- *Lipschitz score, i.e.* $\exists\ L > 0$ *such that* $|\nabla \log p_t(x) - \nabla \log p_t(y)| \leq L|x - y|$.

Theorem 1 (Wasserstein bound on $\mathcal{W}_2^2(p_0, \text{Law}(Y_T))$)**.** *Provided Assumptions 2 and 3 hold, and for hyperparameter $\lambda > 0$,*

$$\begin{aligned} \mathcal{W}_2^2(p_0, Law(Y_T)) \leq (e^{-\alpha_T}\mathbb{E}[\|x_0\|^2] + d(1 - \sqrt{1 - \exp(-\alpha_T)})^2)(e^{(1+2(L+\lambda))\alpha_T}) \\ + \frac{\epsilon^2_{score}}{2\lambda} \int_0^T \beta_t e^{(1+2(L+\lambda))\alpha_t} \mathrm{d}t. \end{aligned} \tag{10}$$

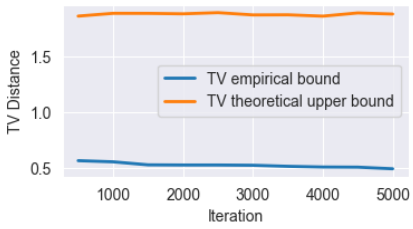
A derivation of the TV bound and proof of Theorem 1 are provided in Appendix B.

Remark 3. *Similar to Kwon et al. [2022], we assume an L^∞ bound on score matching, and if we were to assume instead an L^2 bound, the result still holds as long as the score regularity in Assumption 3 is applied to the learned score instead of the Stein score function. For an L^2 bound on the score matching, see Gao et al. [2025].*

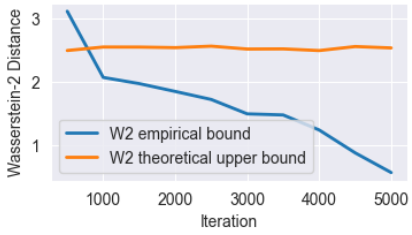
4 Results

In the experiment in this section, we will use the time-dependent OU process in Appendix A. We report (i) empirical Total Variation and Wasserstein-2 bounds between ground truth and generated residuals, (ii) expectations of test functionals against analytic oracles, and (iii) surface plots of the first two dimensions of the ground truth and generated residuals. We compare expectations of test functionals of the residuals (cross moment and squared norms) as an additional distributional check. The results demonstrate that our framework is capable of recovering latent structure in the noise distribution, particularly for multimodal distributions. Details of the experiment are further outlined in Appendix C.

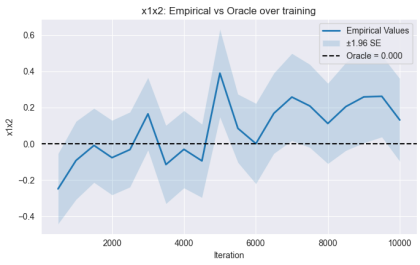
This study focuses on synthetic high-dimensional processes, and the convergence results are derived under strong regularity assumptions. Empirical TV and W_2 distances were upper-bounded by their theoretical bounds, with deviations decreasing over training iterations, suggesting our convergence estimates are informative in practice. The agreement of cross moment and squared norm test functionals with their analytic oracles demonstrates the method preserves essential first and second-order structure. The surface plots confirm that the generated residuals capture the geometry of the ground-truth residuals. Notably, in 20 dimensions, the model successfully recovers multimodal residual distributions. Further work is required to assess robustness as dimensionality grows. Extending the framework to real-world financial data and more complex dynamics remains an important direction.



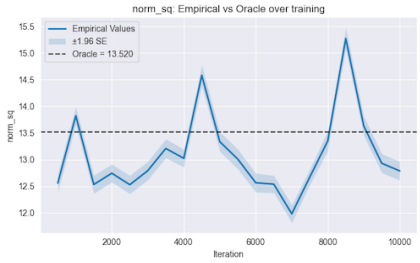
(a) Theoretical Upper Bound (9) vs Empirical TV distance.



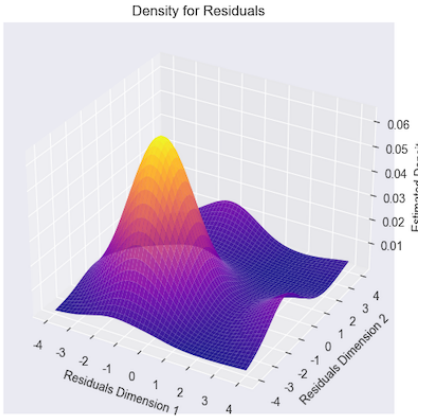
(b) Theoretical Upper Bound (10) vs Empirical W_2 distance.



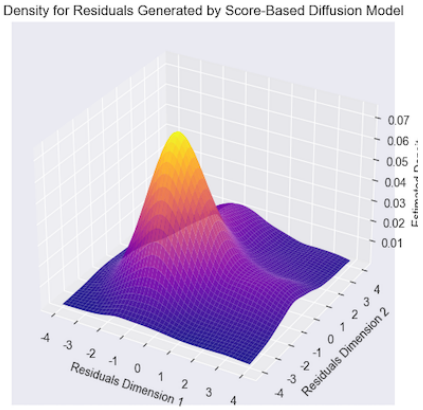
(c) Cross moment for the first two moments of residuals.



(d) Squared norm of components for the first two moments of residuals.



(e) Surface plot of the first 2 dimensions of the residuals.



(f) Surface plot of the first 2 dimensions of the residuals generated by the model.

Figure 1: 1a and 1b are plots for the theoretical versus empirical Total Variation distance and Wasserstein-2 distance. 1c and 1d show expectations of test functionals as targeted probes of the generated samples against analytic oracles computed directly from the (known) data generating process during training, for the cross moment and the squared norm of the moments of the first two dimensions with standard errors, respectively. 1e and 1f show the surface plots of the first two dimensions of ground truth residuals and the residuals generated by the score-based diffusion model.

Acknowledgments

Riya Danait's research is supported by Qube Research and Technologies through the EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation (EPSRC Grant EP/S023925/1).

References

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR, 07–09 Jul 2015.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, and Abhishek Kumar. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representation*, 2021.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. 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.

Milena Vuletić, Felix Prenzel, and Mihai Cucuringu. Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. *Quantitative Finance*, 24(2):175–199, 2024.

Hans Bühler, Blanka Horvath, Terry Lyons, Imanol Perez Arribas, and Ben Wood. A data-driven market simulator for small data environments, 2020.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting, 2023.

Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, Jiang Bian, Shirui Pan, and Qingsong Wen. A survey on diffusion models for time series and spatio-temporal data, 2024.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Chen Su, Zhengzhou Cai, Yuanhe Tian, Zhuochao Chang, Zihong Zheng, and Yan Song. Diffusion models for time series forecasting: A survey, 2025.

Arthur Stéphanovitch, Eddie Aamari, and Clément Levrard. Generalization bounds for score-based generative models: a synthetic proof, 2025.

Arthur Stéphanovitch. Regularity of the score function in generative models, 2025.

Marta Gentiloni-Silveri and Antonio Ocello. Beyond log-concavity and score regularity: Improved convergence bounds for score-based generative models in w2-distance, 2025.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. Kl convergence guarantees for score diffusion models under minimal data assumptions, 2024.

Stanislas Strasman, Antonio Ocello, Claire Boyer, Sylvain Le Corff, and Vincent Lemaire. An analysis of the noise schedule for score-based generative models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964. doi: 10.1137/1109020.

Geoffrey S. Watson. Smooth regression analysis. *Sankhya: The Indian Journal of Statistics Series A*, 26(4):359–372, 1964.

E. A. Nadaraya. Remarks on non-parametric estimates for density functions and regression curves. *Theory of Probability & Its Applications*, 15(1):134–137, 1970. doi: 10.1137/1115015.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17695–17709. Curran Associates, Inc., 2021.

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*, 2023.

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the wasserstein distance. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 20205–20217. Curran Associates, Inc., 2022.

Xuefeng Gao, Hoang M. Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *Journal of Machine Learning Research*, 26(43): 1–54, 2025.

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

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, page 1188–1205, 1986.

Hans Föllmer. An entropy approach to the time reversal of diffusion processes. In *Stochastic Differential Systems Filtering and Control: Proceedings of the IFIP-WG 7/1 Working Conference*, 2005.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106:1602–1614, 12 2011.

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

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

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

A Background on Score-Based Diffusion Models

In Section 1, we introduced the idea of time-reversed diffusions. Below, we state the property for clarity. Consider the following well-defined SDE:

$$\begin{array}{rcl} \mathrm{d}X_t & = & f(X_t,t)\mathrm{d}t + g(X_t,t)\mathrm{d}W_t \\ X_0 & \sim & p_0 \end{array} \tag{11}$$

f and g satisfy local Lipschitz continuity and linear growth conditions, so the existence of p_t is guaranteed. Additionally, p_t is differentiable and strictly positive, provided that $g(x,t)g(x,t)^\top$ is

positive definite. Starting from the density p_T, we expect that running X in reverse time would generate samples from the density p_0. This time reversal property of diffusions is a well-known fact in stochastic analysis (Anderson [1982], Haussmann and Pardoux [1986], Föllmer [2005]).

Proposition 1 (Time Reversal Haussmann and Pardoux [1986])**.** *Consider the SDE 11. Let $Y_t = X_{T-t}$ for $t \in [0, T]$, $T > 0$. Then, under the conditions outlined above, Y is a diffusion process with drift given by*

$$\tilde{f}(x,t) = -f(x, T-t) + \frac{\operatorname{div}(p_{T-t}(x) \cdot a(x, T-t))}{p_{T-t}(x)}, \tag{12}$$

where $a(x,t) = g(x,t)g(x,t)^\top$. Expanding the divergence term component-wise,

$$(\operatorname{div}(p_{T-t}(x) \cdot a(x, T-t)))^i = \sum_{j=1}^{d} \frac{\partial}{\partial x^j}(p_{T-t}(x) a^{ij}(x, T-t)) \tag{13}$$

$$= \sum_{j=1}^{d} \left[\frac{\partial p_{T-t}(x)}{\partial x^j} a^{ij}(x, T-t) + p_{T-t}(x) \frac{\partial a^{ij}(x, T-t)}{\partial x^j} \right], \tag{14}$$

leads to the vector form

$$\operatorname{div}(p_{T-t}(x) \cdot a(x, T-t)) = p_{T-t}(x) \operatorname{div} a(x, T-t) + a(x, T-t) \nabla p_{T-t}(x). \tag{15}$$

Then

$$\tilde{f}(x,t) = -f(x, T-t) + \operatorname{div} a(x, T-t) + a(x, T-t) \nabla \log\left(p_{T-t}(x)\right) \tag{16}$$

satisfying

$$\begin{array}{rcl} \mathrm{d}Y_t & = & \tilde{f}(Y_t, t)\mathrm{d}t + g(Y_t, T-t)\mathrm{d}\bar{W}_t \\ Y_0 & \sim & p_T. \end{array} \tag{17}$$

Running the backward procedure will generate $Y_T \sim p_0$ at time T.

We note a few issues that arise if we want to run the reverse process: we do not have sample access to p_T the initial condition of the reverse SDE, and we do not know p_t, which means we do not know the drift $\nabla \log p_{T-t}$. The easiest way to deal with the initial condition is to consider choosing f and g such that X_t converges to a prior distribution p_∞. This allows the initial distribution of the reverse process to be $Y_0 \sim p_\infty$. We want p_T and p_∞ to be sufficiently close, so that the distribution of X_T is close to p_0. In practice, we choose the parameters so that the distribution p_∞ is Gaussian. Then we only need to compute $\nabla \log p_{T-t}$.

The task of estimating the score function $\nabla \log p_t$ (Ho et al. [2020], Song and Ermon [2019], Song et al. [2021]) is **score matching**, and it involves reducing the estimation of the score function to a supervised learning task. Score matching dates back to Tweedie's Formula from the '50s Efron [2011]. Essentially, we will see that estimating $\nabla \log p_t$ is equivalent to estimating the noise added.

Proposition 2 (Tweedie's Formula)**.** *Given $\tilde{x} = x + e$ for $x \sim p$ and $e \sim \mathcal{N}(0, \sigma^2 \cdot I)$,*

$$\mathbb{E}[x \mid \tilde{x}] = \tilde{x} + \sigma^2 \cdot \nabla \log \tilde{p}(\tilde{x})$$

where $\tilde{p}$ is the density for $\tilde{x}$.

Proof. Since $e \sim \mathcal{N}(0, \sigma^2 I)$, the density of $\tilde{x}$ is:

$$\tilde{p}(\tilde{x}) = \int p(x) \cdot \rho_\sigma(\tilde{x} - x)\, dx, \tag{18}$$

where $\rho_\sigma(z) \propto \exp\left(-\frac{z^2}{2\sigma^2}\right)$ is a Gaussian with variance σ^2. The posterior expectation of x given $\tilde{x}$ is:

$$\mathbb{E}[x \mid \tilde{x}] = \frac{\int x\, p(x)\, \rho_\sigma(\tilde{x} - x)\, dx}{\int p(x)\, \rho_\sigma(\tilde{x} - x)\, dx}. \tag{19}$$

Taking the gradient of $\rho_\sigma(\tilde{x} - x)$ with respect to $\tilde{x}$:

$$\nabla_{\tilde{x}} \rho_\sigma(\tilde{x} - x) = \frac{x - \tilde{x}}{\sigma^2} \rho_\sigma(\tilde{x} - x). \tag{20}$$

Differentiating the log of $\tilde{p}(\tilde{x})$:

$$\nabla_{\tilde{x}} \log \tilde{p}(\tilde{x}) = \frac{\int \frac{x-\tilde{x}}{\sigma^2} p(x) \rho_\sigma(\tilde{x} - x)\, dx}{\int p(x) \rho_\sigma(\tilde{x} - x)\, dx}, \tag{21}$$

which simplifies to:

$$\nabla_{\tilde{x}} \log \tilde{p}(\tilde{x}) = \frac{\mathbb{E}[x \mid \tilde{x}] - \tilde{x}}{\sigma^2}. \tag{22}$$

Rearranging this equation yields Tweedie's formula:

$$\mathbb{E}[x \mid \tilde{x}] = \tilde{x} + \sigma^2 \nabla \log \tilde{p}(\tilde{x}). \tag{23}$$

□

We can consider $\nabla \log \tilde{p}(\tilde{x})$ as the Bayes optimal estimate of the noise – hence given a noisy sample X_t, the supervised learning task is to predict the noise added. In the following definitions, we formalize the concept of score matching. We assume a collection of score estimates $\{s_\theta(x, t)\}$ on $\mathbb{R}^d \times \mathbb{R}_+$ parameterized by θ – typically a neural network. The objective is to solve the following optimization problem:

$$\min_\theta \mathbb{E}_{p_t}[\|\nabla \log p_t(X_t, t) - s_\theta(X_t, t)\|^2]. \tag{24}$$

This is not possible to calculate as we do not know $\nabla \log p_t(X_t, t)$. An alternative approach is that of **implicit score matching**.

Definition 1 (Implicit Score Matching)**.** *Hyvärinen [2005] We compute*

$$\min_\theta \mathbb{E}_{p_t}[\|s_\theta(X_t, t)\|^2 + 2\nabla s_\theta(X_t, t)]. \tag{25}$$

However, implicit score matching may be computationally complex if the dimension d is very large – gradient descent methods would not be efficient as the computation of the gradient of the score network scales linearly in the dimension. The method of **denoising score matching** is one possible approach when working with high-dimensional data.

Definition 2 (Denoising Score Matching)**.** *Vincent [2011] We condition X_t on X_0, replacing $\nabla \log p_t(X_t, t)$ with $\nabla \log p_{t|0}(X_t \mid X_0)$:*

$$\min_\theta \mathbb{E}_{x_0 \sim p_{data}} \mathbb{E}_{x \sim p_{t|0}(x|x_0)}[\|\nabla \log p_{t|0}(x \mid x_0) - s_\theta(x, t)\|^2]. \tag{26}$$

To show the equivalence between 24 and 26, we start with the standard objective, expanding the squared norm:

$$\begin{aligned}\mathbb{E}_{p_t}\left[\|\nabla \log p_t(X_t) - s_\theta(X_t, t)\|^2\right] = \mathbb{E}_{p_t}\left[\|\nabla \log p_t(X_t)\|^2\right] - 2\,\mathbb{E}_{p_t}\left[\langle \nabla \log p_t(X_t), s_\theta(X_t, t)\rangle\right] \\ + \mathbb{E}_{p_t}\left[\|s_\theta(X_t, t)\|^2\right].\end{aligned} \tag{27}$$

Now, we note that the marginal score in the cross-term can be replaced by the conditional score:

$$\mathbb{E}_{p_t}\left[\langle \nabla \log p_t(X_t), s_\theta(X_t, t)\rangle\right] = \mathbb{E}_{x_0 \sim p_0} \mathbb{E}_{x \sim p_{t|0}(x|x_0)}\left[\langle \nabla \log p_{t|0}(x \mid x_0), s_\theta(x, t)\rangle\right]. \tag{28}$$

Given that $\mathbb{E}_{p_t}\left[\|\nabla \log p_t(X_t)\|^2\right]$ and $\mathbb{E}_{p_t}\left[\|s_\theta(X_t, t)\|^2\right]$ are both unaffected by the conditioning on X_0 directly, we can rewrite the entire objective incorporating this conditioning:

$$\mathbb{E}_{p_t}\left[\|\nabla \log p_t(X_t) - s_\theta(X_t, t)\|^2\right] = \mathbb{E}_{x_0 \sim p_0} \mathbb{E}_{x \sim p_{t|0}(x|x_0)}\left[\|\nabla \log p_{t|0}(x \mid x_0) - s_\theta(x, t)\|^2\right], \tag{29}$$

which is exactly the denoising score matching objective. To reiterate, the goal of denoising score matching is to show that the score function of some "noisy" sample should move to a clean sample gradually. We saw that the conditional distribution $p_{t|0}(X_t \mid X_0)$ should be something simple, ideally Gaussian.

Recall the example we look at in our numerical experiments is that of a time dependent "variance preserving" OU process, also considered in Song et al. [2021]:

$$\begin{array}{rcl} \mathrm{d}X_t & = & -\frac{1}{2}\beta_t X_t \mathrm{d}t + \sqrt{\beta_t} dW_t \\ X_0 & \sim & p_0. \end{array} \tag{30}$$

where β_t is a time-dependent function. Let us define $\alpha_t = \int_0^t \beta_s \mathrm{d}s$. Then the transition kernel of X is given by

$$p_{t|0}(\cdot \mid X_0 = x_0) = \mathcal{N}(m_t x_0, v_t I), \tag{31}$$

with $m_t = \exp(-\frac{1}{2}\alpha_t)$ and $v_t = 1 - \exp(-\alpha_t)$, since

$$\begin{array}{rcl} X_t & = & e^{-\frac{\alpha_t}{2}} X_0 + \int_0^t e^{-\frac{(\alpha_t - \alpha_s)}{2}} \sqrt{\beta_s} \mathrm{d}B_s \\ \text{Var} & = & \int_0^t e^{-\alpha_t + \alpha_s} \beta_s \mathrm{d}s = \int_0^t e^{-\alpha_t + \alpha_s} \mathrm{d}\alpha_s \\ \text{Var} & = & e^{-\alpha_t} \int_0^t e^{\alpha_s} \mathrm{d}\alpha_s \\ \text{Var} & = & e^{-\alpha_t}(e^{\alpha_t} - 1) = 1 - \exp(-\alpha_t) \end{array} \tag{32}$$

Implementing this SDE with the time-change parameters α_t and β_t directly impacts the performance quite a bit – it is a strategy for controlling the variance of the noise added which affects the rate at which the data distribution is converted to a tractable noise distribution. In this example, we consider the following β_t introduced in Karras et al. [2022]:

$$\beta_t = \beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}}). \tag{33}$$

Finally, the reverse SDE is given by

$$\begin{array}{rcl} \mathrm{d}Y_t & = & \frac{1}{2}\beta_{T-t} Y_t \mathrm{d}t + \beta_{T-t} \nabla \log p_{T-t}(Y_t) \mathrm{d}t + \sqrt{\beta_{T-t}} \mathrm{d}W_t, \\ Y_0 & \sim & \mathcal{N}(m_T x_0, v_T I). \end{array} \tag{34}$$

We note that if β_{max} is large, then $p_\infty \sim \mathcal{N}(0, I)$, which is why this is called a variance preserving SDE. Note that $X_t \overset{d}{=} m_t X_0 + \sqrt{v_t}\epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$. Then the score function simplifies to

$$\nabla \log p_{t|0}(x \mid x_0) = \frac{m_t x_0 - x}{v_t} \overset{d}{=} -\frac{\epsilon}{\sqrt{v_t}}. \tag{35}$$

We can define a score network $-\sqrt{v_t} \cdot s_\theta(X_t, t)$ that then predicts the noise ϵ from the noisy data $X_t \overset{d}{=} m_t X_0 + \sqrt{v_t}\epsilon$. Then the denoising score matching objective becomes

$$\mathbb{E}_{x_0 \sim p_{\text{data}}} \mathbb{E}_{x \sim p_{t|0}} \left[\left\| \frac{m_t X_0 - x}{v_t} - s_\theta(X_t, t) \right\|^2 \right] = \mathbb{E}_{x_0 \sim p_{\text{data}}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[\left\| s_\theta(m_t X_0 + \sqrt{v_t}\epsilon, t) + \frac{\epsilon}{\sqrt{v_t}} \right\|^2 \right]. \tag{36}$$

B Proofs of Convergence

From Assumption 1 and Chen et al. [2023], we have

$$TV(\text{Law}(Y_T), p_0(\cdot)) \leq TV(p(T, \cdot), p_{\text{noise}}(\cdot)) + \epsilon_{\text{score}} \sqrt{\frac{T}{2}}. \tag{37}$$

Recall the time-t transition kernel is given by 31. In order to quantify $TV(p(T,\cdot), p_{\text{noise}}(\cdot))$, we use KL divergence $KL(\mathcal{N}(m_t X_0, v_t I)\|\mathcal{N}(0,I))$ and Pinsker's inequality:

$$\frac{1}{2}\left(\operatorname{Tr}(I^{-1}v_T I) + (0 - m_T X_0)^\top I^{-1}(0 - m_T X_0) - d + \log\left(\frac{\det I}{\det(v_T I)}\right)\right) \tag{38}$$

$$= \frac{1}{2}(v_T d + m_T^2 |X_0|^2 - d - d\log(v_T))) \tag{39}$$

$$= \frac{1}{2}(m_T^2 |X_0|^2 - d(1 - v_T + \log(v_T))) \tag{40}$$

$$= \frac{1}{2}(m_T^2 |X_0|^2 - d(m_T^2 + \log(v_T))) \tag{41}$$

$$\leq \frac{1}{2} m_T^2 X_0^2 \text{ as } T \to \infty. \tag{42}$$

Thus, $\mathbb{E}_{p_0}[KL(\mathcal{N}(m_t X_0, v_t I)\|\mathcal{N}(0,I))] \leq \frac{1}{2} m_T^2 X_0^2$, so that

$$TV(p(T,\cdot), p_{\text{noise}}(\cdot)) \leq \sqrt{\frac{1}{4} m_T^2 \mathbb{E}_{p_0}[|X_0|^2]} \leq m_T \frac{\sqrt{\mathbb{E}_{p_0}[|X_0|^2]}}{2}. \tag{43}$$

Therefore, the complete inequality is

$$TV(\text{Law}(Y_T), p_0(\cdot)) \leq m_T \frac{\sqrt{\mathbb{E}_{p_0}[|X_0|^2]}}{2} + \epsilon_{\text{score}} \sqrt{\frac{T}{2}}. \tag{44}$$

To prove Theorem 1, we proceed by using coupled SDEs and a Grönwall-type argument. We will construct a coupling between A_t, the exact reverse-time diffusion (which uses the true score) and B_t, the approximate reverse-time diffusion (which uses the learned score). Then we can bound the Wasserstein-2 distance by

$$\mathcal{W}_2(p_0, \text{Law}(Y_T))^2 \leq \mathbb{E}[\|A_T - B_T\|^2]. \tag{45}$$

We consider the same Brownian motion W_t and define $A_0 \sim p_T, B_0 \sim p_{\text{noise}}$. We have the following coupled SDEs:

$$\begin{cases} \mathrm{d}A_t = [-f(A_t, T-t) + g^2(T-t)\nabla \log p_{T-t}(A_t)]\mathrm{d}t + g(T-t)\mathrm{d}W_t \\ \mathrm{d}B_t = [-f(B_t, T-t) + g^2(T-t) s_\theta(B_t, T-t)]\mathrm{d}t + g(T-t)\mathrm{d}W_t \end{cases} \tag{46}$$

Define the coupling error by

$$\delta_t := \mathbb{E}[\|A_t - B_t\|^2]. \tag{47}$$

Applying Itó's formula yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta_t = 2\mathbb{E}[(A_t - B_t)(\tilde{f}_A(t) - \tilde{f}_B(t))], \tag{48}$$

where $\tilde{f}_A(t)$ and $\tilde{f}_B(t)$ are the drift coefficients of A_t and B_t, respectively. Decomposing gives us

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta_t = \underbrace{-2\mathbb{E}[(A_t - B_t)(f(A_t, T-t) - f(B_t, T-t))]}_{C_1} \tag{49}$$

$$+ \underbrace{2\mathbb{E}[(A_t - B_t) g^2(T-t)(\nabla \log p_{T-t}(A_t) - s_\theta(B_t, T-t))]}_{C_2}. \tag{50}$$

By Assumption 3, we have

$$C_1 \leq -2\rho(T-t)\delta_t. \tag{51}$$

Next, we again decompose C_2 to get

$$C_2 = 2g^2(T-t)(\mathbb{E}[(A_t - B_t)](\nabla \log p_{T-t}(A_t) - \nabla \log p_{T-t}(B_t)) + \mathbb{E}[(A_t - B_t)](\nabla \log p_{T-t}(B_t) - s_\theta(B_t, T-t))). \tag{52}$$

By Young's inequality and Assumptions 2 and 3, we obtain

$$C_2 \leq 2g^2(T-t)\left(L\delta_t + \lambda\delta_t + \frac{\epsilon^2_{\text{score}}}{4\lambda}\right) \tag{53}$$

for some hyperparameter λ. Therefore,

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta_t \leq [-2\rho(T-t) + 2g^2(T-t)(L+\lambda)]\delta_t + \frac{\epsilon^2_{\textbf{score}}}{2\lambda}g^2(T-t). \tag{54}$$

Then we can define

$$I(t) := \int_{T-t}^{T} [-2\rho(s) + 2g^2(s)(L+\lambda)]\mathrm{d}s, \tag{55}$$

so when we apply Grönwall's inequality, we have

$$\delta_T \leq e^{I(T)}\delta_0 + \frac{\epsilon^2_{\textbf{score}}}{2\lambda}\int_0^T g^2(t)e^{I(T)-I(T-t)}\mathrm{d}t. \tag{56}$$

Finally, we get

$$\mathcal{W}_2(p_0, \text{Law}(Y_T)) \leq \sqrt{\mathcal{W}_2^2(p_T, p_{\text{noise}})e^{I(T)} + \frac{\epsilon^2_{\textbf{score}}}{2\lambda}\int_0^T g^2(t)e^{I(T)-I(T-t)}\mathrm{d}t}. \tag{57}$$

We again can apply the Wasserstein-2 distance to our setup. In particular,

$$I(t) = \int_{T-t}^{T} [-2\rho(s) + 2g^2(s)(L+\lambda)]\mathrm{d}s \tag{58}$$

$$= \int_{T-t}^{T} [\beta_s + 2(L+\lambda)\beta_s]\mathrm{d}s \tag{59}$$

$$= (1 + 2(L+\lambda))\int_{T-t}^{T} \beta_s \mathrm{d}s \tag{60}$$

$$= (1 + 2(L+\lambda))(\alpha_T - \alpha_{T-t}). \tag{61}$$

Thus, $I(T) = (1 + 2(L+\lambda))\alpha_T$. Additionally,

$$\mathcal{W}_2^2(p_T, p_{\text{noise}}) = \mathcal{W}_2^2(\mathcal{N}(m_T x_0, v_T I_d), \mathcal{N}(0, I)) \leq m_T^2 \mathbb{E}[\|x_0\|^2] + d(\sqrt{v_T} - 1)^2. \tag{62}$$

Since $m_T = \exp(-\frac{1}{2}\alpha_T)$ and $v_T = 1 - \exp(-\alpha_T)$, we have

$$\mathcal{W}_2^2(\mathcal{N}(m_T x_0, v_T I_d), \mathcal{N}(0, I)) = \exp(-\alpha_T)\|x_0\|^2 + d\left(1 - \sqrt{1 - \exp(-\alpha_T)}\right)^2. \tag{63}$$

We conclude

$$\mathcal{W}_2^2(p_0, \text{Law}(Y_T)) \leq (e^{-\alpha_T}\mathbb{E}[\|x_0\|^2] + d(1 - \sqrt{1 - \exp(-\alpha_T)})^2)(e^{(1+2(L+\lambda))\alpha_T}) + \frac{\epsilon^2_{\textbf{score}}}{2\lambda}\int_0^T \beta_t e^{(1+2(L+\lambda))\alpha_t}\mathrm{d}t. \tag{64}$$

C Details of Numerical Experiments

In all of the numerical experiments in this section, we will use the time-dependent "variance preserving" OU process from the example in Appendix A. We now assume that we have N samples $\{x^n\}_{n=1}^N$ from our target distribution p_0. The empirical measure

$$\hat{p}_0 = \frac{1}{N}\sum_{n=1}^{N} \delta_{x^n} \tag{65}$$

is an approximation to p_0. If we start the forward SDE in p_0, we get marginals $\hat{p}_t$ defined below, where we apply the transition kernel to each data point in the empirical distribution x^n at time 0 to x_t and then average over all transition probabilities, as the empirical distribution at time t can be approximated by the mean of the distributions resulting from diffusing each of the original N data points according to the process:

$$\hat{p}_t(x_t) = \frac{1}{N}\sum_{n=1}^{N} p_{t|0}(x_t \mid x^n), \tag{66}$$

which is just a Gaussian mixture with N components, one for each sample x^n. The components are centered at $m_t x^n$ and have variance v_t. These empirical marginals can actually be evaluated (unlike the unknown p_t). The reverse SDE is given by 34. We implement it using the Euler-Maruyama scheme. To advance the SDE by Δt, we compute the following iteration:

$$Y_{t_{i+1}} = Y_{t_i} + (t_{i+1} - t_i)\left(\frac{1}{2}\beta_{T-t}Y_{t_i} + \beta_{T-t}\nabla \log p_{T-t}(Y_{t_i})\right) + \sqrt{\beta_{T-t}}Z_{t_{i+1}-t_i}, \tag{67}$$

where $Z_{t_{i+1}-t_i}$ are independent with distribution $Z_{t_{i+1}-t_i} \sim \mathcal{N}(0, Z_{t_{i+1}-t_i}I)$. We will run the forward SDE until time $T = 1$. Then the time interval for the backward SDE is also $[0, T]$. We discretize this time interval into $(t_i)_{i=1}^L$, $t_0 = 0, t_L = 1$ and run the above scheme. We use $L = 1000$ steps of the reverse SDE; in practical applications, we might try to reduce the number of steps. Additionally, we use a geometric noise schedule for β_t:

$$\beta_t = \beta_{\text{max}}^{1-t}\beta_{\text{min}}^{t} = \beta_{\text{max}}\left(\frac{\beta_{\text{min}}}{\beta_{\text{max}}}\right)^{t}. \tag{68}$$

In practice, we discretize over $R = 10$ steps, so that

$$\beta_r = \beta_{\text{max}}\left(\frac{\beta_{\text{min}}}{\beta_{\text{max}}}\right)^{\frac{r}{R-1}}, \tag{69}$$

for $r = 0, \ldots, R-1$. Instead of using the linearly spaced step size $t_{i+1} - t_i$ directly, we define it through $\varepsilon\left(\frac{\beta_{t_i}}{\beta_T}\right)^2$, where $\beta_T = \beta_{R-1}$ (the final step in the schedule). This relative scaling ensures that the step sizes are larger early on and decrease over time, since the magnitude of updates must decay as β_i shrinks.

We can now plug in the empirical drift $\nabla \log \hat{p}_t$ into the reverse SDE and run it. The result is the exact reverse SDE for the data distribution $p_0 = \hat{p}_0$. Recall that we can also exactly recover $\hat{p}_0$. Indeed, since $p_{t,0}$ is Gaussian we can evaluate the gradient as

$$\nabla \log p_{t,0}(x \mid x_0) = \nabla \log\left((2\pi v_t)^{-d/2}\exp\left(-\frac{\|x - m_t x_0\|^2}{2v_t}\right)\right) \tag{70}$$

$$= \nabla\left[-\frac{d}{2}\log(2\pi v_t) - \frac{\|x - m_t x_0\|^2}{2v_t}\right] \tag{71}$$

$$= -\frac{(x - m_t x_0)}{v_t}. \tag{72}$$

Since we do not have access to $\nabla \log \hat{p}_t$, we approximate it using a neural network using 26. The objective is 29, and if we let

$$\bar{L}(\theta, t) = \mathbb{E}_{x_0 \sim \hat{p}_{\text{data}}} \mathbb{E}_{x \sim p_{t|0}(x|x_0)} \left[\|\nabla \log p_{t|0}(x \mid x_0) - s_\theta(x, t)\|^2 \right], \tag{73}$$

then we need to optimize the network for all t, not just one specific t, and therefore use

$$\bar{L}(\theta) = \mathbb{E}_{t \sim U[0,1]}[\bar{L}(\theta, t)]. \tag{74}$$

This loss can now be approximated by randomly choosing data points from the training batch (as samples from $\hat{p}_0$ and also randomly generating times $t \sim \mathcal{U}[0, 1]$).

We focus on the application of this method to synthetic data. In particular, we test a multivariate time series – a vector AR(1) process where a mixture of Gaussians generates the innovations. We define $\phi = \{\phi_1, \phi_2\} \in \mathbb{R}^{d \times d}$ to be the AR coefficient matrix. Then we define $\varepsilon_t \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$ to be the innovations, where $\mu_k \in \mathbb{R}^d$ and $\Sigma_k \in \mathbb{R}^{d \times d}$ are the mean and covariance for each mixture component $k = 1, \ldots, K$. Therefore, each path evolves as $X_t = \phi X_{t-1} + \varepsilon_t$. We simulate data in $d = 20$ dimensions with $T = 1000$.

The score-based diffusion model is a four layer feed-forward network, and it consists of a linear projection with a GELU activation and a learnable embedding layer, followed by a three layer feedforward network with dropout-regularized GELU activations. Optimization is Adam (learning rate 5×10^{-3}), batch size is 128, and training is run for 10000 iterations. Reverse-time sampling uses Euler-Maruyama with step sizes scaled as u_i and $T_{\text{emp}} = 200$ inner steps. It is trained on the filtered residuals using denoising score matching and the *exact* Gaussian conditional target for the marginals.