A REVERSIBLE SOLVER FOR DIFFUSION SDES

Zander W. Blasingame*
Clarkson University

Chen Liu
Clarkson University

ABSTRACT

Diffusion models have quickly become the state-of-the-art for generation tasks across many different data modalities. An important ability of diffusion models is the ability to encode samples from the data distribution back into the sampling prior distribution. This is useful for performing alterations to real data samples along with guided generation via the continuous adjoint equations. We propose an *algebraically reversible* solver for diffusion SDEs that can exactly invert real data samples into the prior distribution.

1 INTRODUCTION

Diffusion models have quickly become the state-of-the-art in many different modalities in generation, *e.g.*, audio (Liu et al., 2023), images (Rombach et al., 2022), video (Blattmann et al., 2023), protein generation (Skreta et al., 2024), *&c*. The sampling process of diffusion models is done through numerically solving an Itô Stochastic Differential Equation (SDE) or related Ordinary Differential Equation (ODE) which describes the evolution of a sample drawn for some prior noise distribution to the data distribution. Inversion of the sampling procedure, *i.e.*, an encoding from the data distribution back to the prior distribution, is invaluable for many downstream applications. *E.g.*, image editing (Hertz et al., 2023; Su et al., 2023; Meng et al., 2022; Nie et al., 2024) and image interpolation (Song et al., 2021a; Blasingame & Liu, 2024a;b) with natural extensions to other data modalities. Existing work on diffusion inversion has focused on the ODE formulation (Wallace et al., 2023; Zhang et al., 2024; Wang et al., 2024); however, recent work (Nie et al., 2024) has shown that the SDE formulation is particularly useful for applications when the latent representation is edited. Motivated by this finding, we propose a novel *algebraically reversible* solver for diffusion SDEs that makes use of the Brownian interval (Kidger et al., 2021) to perform *exact* inversion with diffusion models without storing the entire Wiener process in memory. To the best of our knowledge this work is the *first* to propose a technique for exactly inverting diffusion SDEs *without* storing the sampled noise from every timestep in memory.

2 PRELIMINARIES

Diffusion models aim to learn a mapping from some simple prior distribution of Gaussian noise $p(\boldsymbol{x})$ to the data distribution $q(\boldsymbol{x})$. The namesake for this class of models comes from the forward diffusion process, an Itô SDE given by:

$$\mathrm{d}\boldsymbol{X}_t = f(t)\boldsymbol{X}_t + g(t)\,\mathrm{d}\boldsymbol{W}_t, \tag{2.1}$$

where $f, g \in \mathcal{C}^1([0,T])$ form the drift and diffusion coefficients of the SDE, and where $\{\boldsymbol{W}_t\}_{t\in[0,T]}$ is the standard Wiener process on the time interval $[0,T]$. The *reverse-time* SDE (Anderson, 1982) of Equation (2.1) is found to be:

$$\mathrm{d}\boldsymbol{X}_t = [f(t)\boldsymbol{X}_t + g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{X}_t)]\,\mathrm{d}t + g(t)\,\mathrm{d}\overline{\boldsymbol{W}}_t, \tag{2.2}$$

where $\mathrm{d}t$ is a *negative* timestep and $\{\overline{\boldsymbol{W}}_t\}_{t\in[0,T]}$ is the standard Wiener process in reverse-time. The aim of diffusion models, then, is to learn the score function $\boldsymbol{s}_\theta(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ (Song et al., 2021b) or a closely related quantity, *e.g.*, noise prediction (Song et al., 2021a; Ho et al., 2020) or data prediction (Kingma et al., 2021). Once the score function or another equivalent parameterization is learned, we can use it to sample $q(\boldsymbol{x})$ by first sampling some $\boldsymbol{x}_T \sim p(\boldsymbol{x})$ and then numerically solving Equation (2.2) in *reverse-time* with our model of the score function with some kind of SDE solver, *e.g.*, Euler-Maruyama.

*Correspondance to `blasinzw@clarkson.edu`

3 MOTIVATION

A common task with diffusion models is to encode samples from the data distribution $q(\boldsymbol{x})$ back into the noise distribution $p(\boldsymbol{x})$. This can be for tasks such as image editing (Meng et al., 2022; Nie et al., 2024), or for performing the backward pass in solving the continuous adjoint equations (Blasingame & Liu, 2024a; Pan et al., 2024). However, simply solving the diffusion ODE forwards in time[1] can raise some issues due to truncation errors and stability concerns. Furthermore, integrating the diffusion SDE in forwards time requires a bit more care due to the Itô stochastic integral.

Truncation errors. Suppose that we have some numerical scheme for sampling the diffusion model, *e.g.*, DDIM (Song et al., 2021a), DEIS (Zhang & Chen, 2023), or DPM-Solver (Lu et al., 2022a;b); such that we can sample the solution trajectory $\{\tilde{\boldsymbol{x}}_{t_n}\}_{n=0}^N$ with timesteps $\{t_N = T > t_{N-1} \cdots > t_0 = 0\}$ and where $\tilde{\boldsymbol{x}}_{t_N} \sim q(\boldsymbol{x})$ and the rest of the trajectory is found via the numerical scheme. Now, suppose we apply the same scheme forwards in time with initial condition $\check{\boldsymbol{x}}_{t_0} = \tilde{\boldsymbol{x}}_{t_0}$ to construct the encoding trajectory $\{\check{\boldsymbol{x}}_{t_n}\}_{n=1}^N$, we have no guarantee that encoding trajectory will equal the sampling trajectory for $n > 0$, *i.e.*, $\check{\boldsymbol{x}}_{t_n} = \tilde{\boldsymbol{x}}_{t_n}$ does not *necessarily* hold for $n > 0$. This has a few implications: a) a scheme which is used for encoding and then sampling is not guaranteed to have exact inversion and b) the samples generated in the encoding trajectory differ from the samples in the sampling trajectory, causing inaccurate gradients when using the continuous adjoint equations.

Stability. Similarly, there are concerns about the numerical stability of the numerical scheme solved in both directions of time. Consider the test ODE $\dot{y}(t) = \lambda y(t)$ with $\lambda < 0$ defined in the interval $[0, T]$ with the initial condition y_0. An ODE solver with a nontrivial region of convergence (see Harier & Wanner, 2002) will be able to solve this ODE without much trouble, as the magnitude of the errors decreases exponentially since λ is negative. However, the backwards in time solve from $y(T)$ will suffer numerical instability as the errors will *grow exponentially*. *N.B.*, this problem is simply reversed if the solver has good stability in reverse-time, with the solve in forward-time now suffering. Furthermore, the poor stability in the backward solve is an issue for diffusion guidance techniques which use continuous adjoint equations.

As such we desire a numerical solver for diffusion models which has *alignment* in truncation errors in both directions of time along with reasonable numerical stability in both directions.

4 REVERSIBLE SOLVERS FOR DIFFUSION SDES

Now as we alluded earlier, there are some difficulties with solving Equation (2.2) in forward-time as the Itô integral is adapted to the backward filtration induced by $\{\overleftarrow{\boldsymbol{W}}_t\}$. Instead, it is much easier to use the *Fisk-Stratonovich symmetric integral* (see Kunita, 2019), which has the nice property of symmetry in time. Now as the Itô integral term in Equation (2.2) is only additive noise, we can freely switch to the Stratonovich integral without consequence, thereby rewriting Equation (2.2) as an integral equation of the form:

$$\boldsymbol{X}_t = \int_T^t f(\tau)\boldsymbol{X}_\tau + g^2(\tau)\nabla_{\boldsymbol{x}} \log p_\tau(\boldsymbol{X}_\tau)\, \mathrm{d}\tau + \int_T^t g(\tau) \circ \mathrm{d}\boldsymbol{W}_\tau. \tag{4.1}$$

N.B., the Wiener process $\{\boldsymbol{W}_t\}$ is not the same process as the one used in Equation (2.1), but we adopt it for a simpler notation, that we ask the reader to keep this in mind.[2] The drift and diffusion coefficients, (f, g), are defined via the schedule (α_t, σ_t) in the Variance Preserving (VP) formulation (Song et al., 2021b) with $f(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}, g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2$.

Let $\boldsymbol{x}_{0|t}(\boldsymbol{x}) \coloneqq \mathbb{E}[\boldsymbol{X}_0|\boldsymbol{X}_t = \boldsymbol{x}]$ be the data prediction model. Following Lu et al. (2022a) we let $\lambda \coloneqq \log\alpha_t/\sigma_t$ denote one-half the log-SNR (Signal to Noise Ratio). Since λ_t is a strictly decreasing function of t, there exists an inverse function $t_\lambda(\cdot)$ such that $t = t_\lambda(\lambda_t)$ and, with abuse of notation, we let $\boldsymbol{x}_\lambda \coloneqq \boldsymbol{x}_{t_\lambda(\lambda)}$ and $\boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{x}_\lambda) \coloneqq \boldsymbol{x}_{0|t_\lambda(\lambda)}(\boldsymbol{x}_{t_\lambda(\lambda)})$. We can then simplify the integral equation in Equation (4.1) using *exponential integrators* (Hochbruck & Ostermann, 2010)—a

[1] *N.B.*, due to the time conventions with diffusion models, sampling is performed backwards in time and encoding is forwards in time.

[2] More technically, we define the Stratonovich integral w.r.t. to the natural *two-sided* filtration induced by $\{\boldsymbol{W}_t\}$ defined in forward-time. For more details we refer to Kunita (2019) and for an application in a modern ML context we recommend Li et al. (2020).

common technique for diffusion models (see Lu et al., 2022a; Zhang & Chen, 2023; Gonzalez et al., 2023; Blasingame & Liu, 2024a)—to simplify the solution of the Stratonovich integral equation. This result is presented in Proposition 4.1 with the full proof in Appendix B.

Proposition 4.1 (Exact solution of diffusion SDEs)**.** *Given an initial value $\boldsymbol{X}_s(\omega) = \boldsymbol{x}_s$ at time $s \in [0, T]$ the exact solution of Equation (4.1) can be expressed as:*

$$\boldsymbol{X}_t = \underbrace{\frac{\sigma_t}{\sigma_s} e^{\lambda_s - \lambda_t} \boldsymbol{X}_s}_{\substack{\textbf{\textit{Linear term}} \\ \textit{No truncation errors}}} + \underbrace{2\alpha_t \int_{\lambda_s}^{\lambda_t} e^{2(\lambda - \lambda_t)} \boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\, \mathrm{d}\lambda}_{\substack{\textbf{\textit{Approximated term}} \\ \textit{Truncation errors}}} + \underbrace{\sqrt{2}\sigma_t e^{-\lambda_t} \boldsymbol{W}_{\varsigma_t, \varsigma_s}}_{\substack{\textbf{\textit{Brownian bridge}} \\ \textit{No truncation errors}}}, \tag{4.2}$$

where $\varsigma_t = \frac{1}{2}(e^{2\lambda_t} - e^{2\lambda_T})$.

Proof sketch. First we simplify the integral equation using the method*variation-of-parameters* to get an *exponential integrator* (Hochbruck & Ostermann, 2010). We express the Stratonovich integral as a continuous local martingale in backward time, enabling us to use the Dubins-Schwarz theorem (Dubins & Schwarz, 1965) to rewrite the integral as a time-changed Brownian motion.

The exponential integral term in Equation (4.2) can be approximated via a truncated Stratonovich Taylor expansion (Kloeden & Platen, 1991) and the time-changed Brownian bridge can be efficiently calculated by using the Brownian interval (Kidger et al., 2021, Algorithm 3) in **both** reverse-time and forward-time.

4.1 Construction of the Reversible Solver

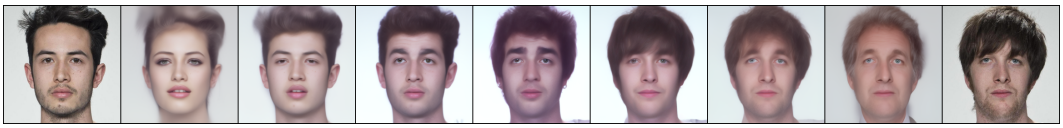
Now equipped with a simplified form of the diffusion SDE we develop a reversible solver based on approximations of the exponential integral in Equation (4.2). Taking inspiration from the recent work of McCallum & Foster (2024), in which they design reversible solvers for neural ODEs with a non-trivial region of convergence, we apply their insight of using a coupling parameter to construct an *algebraically reversible* solver to Equation (4.2). Note that we assume that our data prediction model is trained to zero loss, *i.e.*, $\hat{\boldsymbol{x}}_{0|t} = \boldsymbol{x}_{0|t}$.

Forward pass. Suppose that we have a single-step solver for the exponential integral term in Equation (4.2) given by $\Psi_h : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ where h denotes the step size $h \coloneqq \lambda_t - \lambda_s$ and timesteps $\{t_n\}_{n=1}^N$ which is defined in *reverse-time*. Let $\zeta \in (0, 1)$ be a coupling parameter that determines the stability of the forward and backward passes, and let $\hat{\boldsymbol{x}}$ be an augmented state for algebraic reversibility. For notational simplicity, let $\boldsymbol{x}_n \coloneqq \boldsymbol{x}_{t_n}$ and likewise for other variables. We then define the forward pass as

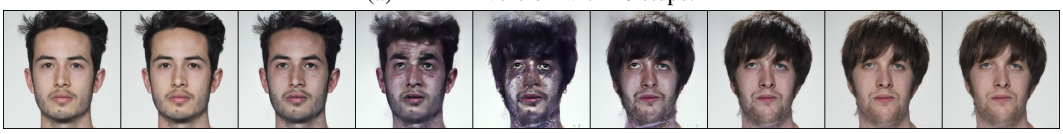
$$\begin{aligned}
\boldsymbol{x}_{n+1} &= \zeta \boldsymbol{x}_n + (1 - \zeta)\hat{\boldsymbol{x}}_n + \frac{\sigma_{n+1}}{\sigma_n} e^{-h} \hat{\boldsymbol{x}}_n + 2\alpha_{n+1} \Psi_h(t_n, \hat{\boldsymbol{x}}_n) \\
&\quad + \sqrt{2}\sigma_{n+1} e^{-\lambda_{n+1}} \boldsymbol{W}_{\varsigma_{n+1}, \varsigma_n}, \\
\hat{\boldsymbol{x}}_{n+1} &= \hat{\boldsymbol{x}}_n - \frac{\sigma_n}{\sigma_{n+1}} e^{h} \boldsymbol{x}_{n+1} - 2\alpha_n \Psi_{-h}(t_{n+1}, \boldsymbol{x}_{n+1}) + \sqrt{2}\sigma_n e^{-\lambda_n} \boldsymbol{W}_{\varsigma_{n+1}, \varsigma_n}.
\end{aligned} \tag{4.3}$$

Backward pass. The backward solve can then be computed algebraically from Equation (4.3) as

$$\begin{aligned}
\hat{\boldsymbol{x}}_n &= \hat{\boldsymbol{x}}_{n+1} + \frac{\sigma_n}{\sigma_{n+1}} e^{h} \boldsymbol{x}_{n+1} + 2\alpha_n \Psi_{-h}(t_{n+1}, \boldsymbol{x}_{n+1}) - \sqrt{2}\sigma_n e^{-\lambda_n} \boldsymbol{W}_{\varsigma_{n+1}, \varsigma_n}, \\
\boldsymbol{x}_n &= \zeta^{-1} \boldsymbol{x}_{n+1} + (1 - \zeta^{-1})\hat{\boldsymbol{x}}_n - \frac{\sigma_{n+1}}{\sigma_n} e^{-h} \zeta^{-1} \hat{\boldsymbol{x}}_n + 2\alpha_{n+1} \zeta^{-1} \Psi_h(t_n, \hat{\boldsymbol{x}}_n) \\
&\quad - \sqrt{2}\sigma_{n+1} e^{-\lambda_{n+1}} \zeta^{-1} \boldsymbol{W}_{\varsigma_{n+1}, \varsigma_n}.
\end{aligned} \tag{4.4}$$



(a) DDIM inversion with 20 steps.



(b) Reversible DDIM with 20 steps.

(c) Reversible diffusion SDE with 20 steps.

Figure 1: Comparison of different solvers for diffusion models on an image interpolation task with interpolation values $0, 0.1, 0.25, 0.5, 0.75, 0.9, 1$ (from left to right). The leftmost and rightmost images are the original images: $\boldsymbol{x}_0^{(a)}$ and $\boldsymbol{x}_0^{(b)}$. The same number of steps are used for both the encoding and sampling procedure. For the reversible methods $\zeta = 0.999$. Original faces from FRLL (DeBruine & Jones, 2017).

5 Interpolation Experiment

Following Song et al. (2021a) we test our method with a small experiment of image interpolation, *i.e.*, given two real images $\boldsymbol{x}_0^{(a)}$ and $\boldsymbol{x}_0^{(b)}$ they are inverted to find $\boldsymbol{x}_T^{(a)}$ and $\boldsymbol{x}_T^{(b)}$. These representations are then interpolated via spherical linear interpolation (Shoemake, 1985) to obtain an interpolated latent representation. For the reversible solvers we also interpolate $\hat{\boldsymbol{x}}$. For this experiment, we used a Latent Diffusion Model (LDM) (Rombach et al., 2022) trained on the CelebA-HQ dataset (Karras et al., 2018) at a 256×256 resolution. More details can be found in Appendix C.

In Figure 1 we plot an example interpolation with three different solvers: a) the standard DDIM inversion which serves as the baseline, b) McCallum and Foster's method applied to DDIM, and c) the reversible solver described in Equations (4.3) and (4.4) with first-order truncated Stratonovich-Taylor expansion. Notice how the DDIM solver struggles to accurately invert the original images; whereas the same solver when used in conjunction with McCallum and Foster's method yields *exact* inversion. The poor performance of the DDIM inversion is not surprising, as we used a low number of discretization steps to stress test the solvers. The SDE solver with only 20 sampling steps performs better than the ODE-based inversion with fewer visible distortions, while still achieving exact inversion.

6 Conclusion

In this work, we propose a novel *algebraically reversible* solver for diffusion SDEs allowing the exact inversion of samples into the latent space with diffusion SDEs. To the best of our knowledge, we are the first to propose a method to exactly invert diffusion SDEs that *does not* store the entire realization of the Wiener process in memory. We illustrate the utility of our solver on the experiment of image interpolation. This work has many potential applications in the editing of samples with diffusion SDEs and for eliminating truncation errors in guided generation methods which use the continuous adjoint equations.

Limitations. As preliminary work we have only explored a proof of concept experiment with image interpolation to illustrate the stability under perturbations and exact inversion of our method. In the future, we plan to explore using this solver for guided generation via the continuous adjoint equations and perform a more detailed analysis of the stability.

REFERENCES

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

Zander W. Blasingame and Chen Liu. AdjointDEIS: Efficient gradients for diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 2449–2483. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/04badd3b048315c8c3a0ca17eff723d7-Paper-Conference.pdf.

Zander W. Blasingame and Chen Liu. Greedy-dim: Greedy algorithms for unreasonably effective face morphs. In *2024 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–11, 2024b. doi: 10.1109/IJCB62174.2024.10744517.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Lisa DeBruine and Benedict Jones. Face Research Lab London Set. 5 2017. doi: 10.6084/m9.figshare.5047666.v5. URL https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666.

Lester E Dubins and Gideon Schwarz. On continuous martingales. *Proceedings of the National Academy of Sciences*, 53(5):913–916, 1965.

Martin Gonzalez, Nelson Fernandez, Thuy Vinh Dinh Tran, Elies Gherbi, Hatem Hajri, and Nader Masmoudi. SEEDS: Exponential SDE solvers for fast high-quality sampling from diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=V6IgkYKD8P.

Ernst Harier and Gerhard Wanner. *Solving Ordinary Differential Equations II Stiff and Differential-Algebraic Problems.* Springer, Berlin, 2002. Second Revised Edition.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_CDixzkzeyb.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Patrick Kidger, James Foster, Xuechen Chen Li, and Terry Lyons. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Peter E Kloeden and Eckhard Platen. Stratonovich and itô stochastic taylor expansions. *Mathematische Nachrichten*, 151(1):33–50, 1991.

Hiroshi Kunita. *Stochastic Flows and Jump-Diffusions*, volume 92. Springer, 2019.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, pp. 21450–21474. PMLR, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Sam McCallum and James Foster. Efficient, accurate and stable gradients for neural odes. *arXiv preprint arXiv:2410.11648*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: SDE beats ODE in general diffusion-based image editing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DesYwmUG00.

Jiachun Pan, Jun Hao Liew, Vincent Tan, Jiashi Feng, and Hanshu Yan. AdjointDPM: Adjoint sensitivity method for gradient backpropagation of diffusion probabilistic models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y33lDRBgWI.

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

Lawrence F Shampine. Stability of the leapfrog/midpoint method. *Applied mathematics and computation*, 208(1):293–298, 2009.

Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 245–254, 1985.

Marta Skreta, Lazar Atanackovic, Avishek Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the Itô density estimator. *arXiv preprint arXiv:2412.17762*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5HLoTvVGDe.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22532–22541, 2023.

Fangyikang Wang, Hubery Yin, Yue-Jiang Dong, Huminhao Zhu, Chao Zhang, Hanbin Zhao, Hui Qian, and Chen Li. BELM: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ccQ4fmwLDb.

Chen Henry Wu and Fernando De la Torre. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. *arXiv preprint arXiv:2210.05559*, 2022.

Guoqiang Zhang, J. P. Lewis, and W. Bastiaan Kleijn. Exact diffusion inversion via bidirectional integration approximation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LVII*, pp. 19–36, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72997-3. doi: 10.1007/978-3-031-72998-0_2. URL https://doi.org/10.1007/978-3-031-72998-0_2.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

Juntang Zhuang, Nicha C Dvornek, sekhar tatikonda, and James s Duncan. {MALI}: A memory efficient and reverse accurate integrator for neural {ode}s. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=blfSjHeFM_e.

A Related Works

In this section we provide a small discussion on related work.

Reversible solvers. The asynchronous leapfrog method (Zhuang et al., 2021) and the reversible Heun (Kidger et al., 2021) were the standard reversible solvers until recently (*cf*. McCallum & Foster, 2024), with the former applicable to general neural ODEs and the latter applicable to neural ODEs, CDEs and SDEs. Recent work by McCallum & Foster (2024) has improved upon these older solvers for neural ODEs by showing it is possible to construct reversible solvers with a non-trivial region of stability.

Exact inversion with diffusion models. The work of Wallace et al. (2023) proposes a reversible solver for diffusion ODEs by solving a dual auxiliary state of the model and then interpolating between the two states. Later work by Zhang et al. (2024) explores a reversible solver by using a bidirectional integration approximation scheme as a sort of leapfrog method. More recent work by Wang et al. (2024) has explored the exact inversion of diffusion ODEs using bidirectional linear multistep methods. However, linear multi-step methods and leapfrog methods often suffer from poor stability (Shampine, 2009).

Inversion with diffusion SDEs. More closely related to our work Wu & De la Torre (2022) propose a method for the exact inversion of diffusion SDEs. Given a particular realization of the Wiener process that admits $\boldsymbol{x}_t \sim \mathcal{N}(\alpha_t \boldsymbol{x}_0 \mid \sigma_t^2 \mathbf{I})$, then given $\boldsymbol{x}_s$ and noise $\boldsymbol{\epsilon}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ we can calculate

$$\boldsymbol{x}_t = \frac{\alpha_t}{\alpha_s}\boldsymbol{x}_s + 2\sigma_t(e^h - 1)\hat{\boldsymbol{x}}_{T|s}(\boldsymbol{x}_s) + \sigma_t\sqrt{e^{2h} - 1}\boldsymbol{\epsilon}_s. \tag{A.1}$$

Wu & De la Torre (2022) propose to invert this by first calculating for two samples $\boldsymbol{x}_t$ and $\boldsymbol{x}_s$ the noise $\boldsymbol{\epsilon}_s$ can be calculated by rearranging the previous equation to find

$$\boldsymbol{\epsilon}_s = \frac{\boldsymbol{x}_t - \frac{\alpha_t}{\alpha_s}\boldsymbol{x}_s + 2\sigma_t(e^h - 1)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_s, \boldsymbol{z}, s)}{\sigma_t\sqrt{e^{2h} - 1}} \tag{A.2}$$

With this the sequence $\{\boldsymbol{\epsilon}_{t_i}\}_{i=1}^N$ of added noises can be calculated which can be used to reconstruct the original input from the initial realization of the Wiener process. However, unlike our approach, this process requires storing the entire realization in memory.

B Proof of Proposition 4.1

We restate Proposition 4.1 here:

Proposition B.1 (Exact solution of diffusion SDEs)**.** *Given an initial value $\boldsymbol{X}_s(\omega) = \boldsymbol{x}_s$ at time $s \in [0, T]$ the exact solution of Equation (4.1) can be expressed as:*

$$\boldsymbol{X}_t = \underbrace{\frac{\sigma_t}{\sigma_s}e^{\lambda_s - \lambda_t}\boldsymbol{X}_s}_{\substack{\textbf{\textit{Linear term}} \\ \textit{No truncation errors}}} + \underbrace{2\alpha_t \int_{\lambda_s}^{\lambda_t} e^{2(\lambda - \lambda_t)}\boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\,\mathrm{d}\lambda}_{\substack{\textbf{\textit{Approximated term}} \\ \textit{Truncation errors}}} + \underbrace{\sqrt{2}\sigma_t e^{-\lambda_t}\boldsymbol{W}_{\varsigma_t, \varsigma_s}}_{\substack{\textbf{\textit{Brownian bridge}} \\ \textit{No truncation errors}}}, \tag{B.1}$$

where $\varsigma_t = \frac{1}{2}(e^{2\lambda_t} - e^{2\lambda_T})$.

We also restate the Dubins-Schwarz representation theorem (Dubins & Schwarz, 1965) (sometimes referred to as the Dambis representation theorem) below:

Theorem B.2 (Dubins-Schwarz representation theorem)**.** *Let M be a continuous local martingale adapted to a filtration $\{\mathcal{F}_t\}_{t\in[0,\infty)}$ vanishing at zero that satisfies $\langle M\rangle_\infty = \infty$ almost surely. Define the stopping times $\{\tau_t\}_{t\in[0,\infty)}$ by*

$$\tau_t = \inf\{s \in [0, \infty) : \langle M\rangle_s > t\} = \sup\{s \in [0, \infty) : \langle M\rangle_s = t\}.$$

Then, $\{M_{\tau_t}\}_{t\in[0,\infty)}$ is a standard Brownian motion $\{B_t\}_{t\in[0,\infty)}$ and, for every $t \in [0, \infty)$,

$$M_t = B_{\langle M\rangle_t}.$$

Now we state our proof for Proposition 4.1.

Proof. First we restate our Stratonovich integral equation from Equation (4.1):

$$\boldsymbol{X}_t = \boldsymbol{X}_s + \int_s^t f(\tau)\boldsymbol{X}_\tau + g^2(\tau)\nabla_{\boldsymbol{x}} \log p_\tau(\boldsymbol{X}_\tau)\,\mathrm{d}\tau + \int_s^t g(\tau) \circ \mathrm{d}\boldsymbol{W}_\tau, \tag{B.2}$$

where

$$f(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}, \tag{B.3}$$

$$g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2 = -2\sigma_t^2\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}. \tag{B.4}$$

We can express the score function in terms of the data prediction model $\boldsymbol{x}_{0|t}(\boldsymbol{x}) \coloneqq \mathbb{E}[\boldsymbol{X}_0|\boldsymbol{X}_t = \boldsymbol{x}]$:

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = \frac{1}{\sigma_t^2}\boldsymbol{x} - \frac{\alpha_t}{\sigma_t^2}\boldsymbol{x}_{0|t}(\boldsymbol{x}). \tag{B.5}$$

Using Equation (B.5) we can rewrite Equation (4.1) as

$$\boldsymbol{X}_t = \boldsymbol{X}_s + \int_s^t \left(f(\tau) + \frac{g^2(\tau)}{\sigma_\tau^2}\right)\boldsymbol{X}_\tau - \frac{\alpha_\tau g^2(\tau)}{\sigma_\tau^2}\boldsymbol{x}_{0|t}(\boldsymbol{X}_\tau)\,\mathrm{d}\tau + \int_s^t g(\tau) \circ \mathrm{d}\boldsymbol{W}_\tau. \tag{B.6}$$

Then we can write the Stratonovich integral equation as

$$\boldsymbol{X}_t = \boldsymbol{X}_s + \int_s^t a(\tau)\boldsymbol{X}_\tau + b(\tau)\boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\,\mathrm{d}\tau + \int_s^t g(\tau) \circ \mathrm{d}\boldsymbol{W}_\tau, \tag{B.7}$$

where

$$a(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t} + \frac{g^2(t)}{\sigma_t^2} = \frac{1}{\sigma_t^2}\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}, \tag{B.8}$$

$$b(t) = -\frac{\alpha_t g^2(t)}{\sigma_t^2} = 2\alpha_t\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}. \tag{B.9}$$

Then we can use the *variation-of-parameters* formula to find

$$\boldsymbol{X}_t = \Phi_a(s,t)\boldsymbol{X}_s + \int_s^t \Phi_a(\tau,t)b(\tau)\boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\,\mathrm{d}\tau + \int_s^t \Phi_a(\tau,t)g(\tau) \circ \mathrm{d}\boldsymbol{W}_\tau, \tag{B.10}$$

where $\Phi_a(s,t) \coloneqq \exp\int_s^t a(\tau)\,\mathrm{d}\tau$ is the integrating factor. This technique has been employed by other works to separate the linear and non-linear component of diffusion models (Lu et al., 2022a; Gonzalez et al., 2023; Blasingame & Liu, 2024a). We then simplify $\Phi_a(s,t)$ such that

$$\begin{aligned}
\Phi_a(s,t) &= \exp\left(\int_s^t \frac{1}{\sigma_\tau^2}\frac{\mathrm{d}\sigma_\tau^2}{\mathrm{d}\tau}\,\mathrm{d}\tau - \int_s^t \frac{\mathrm{d}\log\alpha_\tau}{\mathrm{d}\tau}\,\mathrm{d}\tau\right),\\
&= \exp\left(\int_{\sigma_s}^{\sigma_t} \frac{1}{\sigma^2}\,\mathrm{d}\sigma^2 - \int_s^t \mathrm{d}\log\alpha_\tau\right),\\
&= \exp\left(\log\sigma_t^2 - \log\sigma_s^2 - (\log\alpha_t - \log\alpha_s)\right),\\
&= \exp\left(\log\frac{\sigma_t^2}{\sigma_s^2} - \log\frac{\alpha_t}{\alpha_s}\right),\\
&= \exp\left(\log\frac{\sigma_t^2\alpha_s}{\sigma_s^2\alpha_t}\right),\\
&= \frac{\sigma_t^2\alpha_s}{\sigma_s^2\alpha_t}.
\end{aligned} \tag{B.11}$$

Another useful form to express the integrating factor in, is in terms of λ_t, which we find:

$$\begin{aligned}
\Phi_a(s,t) &= \frac{\sigma_t^2\alpha_s}{\sigma_s^2\alpha_t},\\
&= \frac{\sigma_t}{\sigma_s}\exp\left(\log\frac{\sigma_t\alpha_s}{\sigma_s\alpha_t}\right),\\
&= \frac{\sigma_t}{\sigma_s}\exp\left(\log\frac{\alpha_s}{\sigma_s} - \log\frac{\alpha_t}{\sigma_t}\right),\\
&= \frac{\sigma_t}{\sigma_s}e^{\lambda_s - \lambda_t}.
\end{aligned} \tag{B.12}$$

Using this expression for $\Phi_a(s,t)$ we can rewrite Equation (B.10) as:

$$\boldsymbol{X}_t = \frac{\sigma_t}{\sigma_s} e^{\lambda_s - \lambda_t} \boldsymbol{X}_s + 2\sigma_t \int_s^t e^{\lambda_\tau - \lambda_t} \frac{\alpha_\tau}{\sigma_\tau} \frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau} \boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\, \mathrm{d}\tau + \sqrt{2}\sigma_t \int_s^t e^{\lambda_\tau - \lambda_t} \sqrt{-\frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau}} \circ \mathrm{d}\boldsymbol{W}_\tau. \tag{B.13}$$

We first simplify the integral term

$$\begin{aligned}
2\sigma_t \int_s^t e^{\lambda_\tau - \lambda_t} \frac{\alpha_\tau}{\sigma_\tau} \frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau} \boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\, \mathrm{d}\tau &= 2\sigma_t \int_s^t e^{\lambda_\tau - \lambda_t} e^{\lambda_\tau} \frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau} \boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\, \mathrm{d}\tau, \\
&= 2\sigma_t \int_s^t e^{2\lambda_\tau - \lambda_t} \frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau} \boldsymbol{x}_{0|\tau}(\boldsymbol{X}_\tau)\, \mathrm{d}\tau, \\
&= 2\sigma_t \int_{\lambda_s}^{\lambda_t} e^{2\lambda - \lambda_t} \boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\, \mathrm{d}\lambda, \\
&= 2\alpha_t \int_{\lambda_s}^{\lambda_t} e^{2(\lambda - \lambda_t)} \boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\, \mathrm{d}\lambda. && \text{(B.14)} \\
&= 2\frac{\sigma_t^2}{\alpha_t} \int_{\lambda_s}^{\lambda_t} e^{2\lambda} \boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\, \mathrm{d}\lambda. && \text{(B.15)}
\end{aligned}$$

To simplify the stochastic integral term we first define a continuous martingale $\boldsymbol{M}_t$ via the stochastic integral:

$$\boldsymbol{M}_t \coloneqq -\int_T^t e^{\lambda_\tau} \sqrt{-\frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau}} \circ \mathrm{d}\boldsymbol{W}_\tau. \tag{B.16}$$

We choose time T as our starting point for the martingale rather than 0 and then integrate in *reverse-time*, hence the negative sign. We can then express our stochastic integral in Equation (B.13) as

$$\int_s^t e^{\lambda_\tau} \sqrt{-\frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau}} \circ \mathrm{d}\boldsymbol{W}_\tau = \boldsymbol{M}_t - \boldsymbol{M}_s. \tag{B.17}$$

Next we establish a few properties of this martingale. First, $\boldsymbol{M}_0 = 0$ by construction. Second, the quadratic variation of $\boldsymbol{M}_t$ is found to be

$$\begin{aligned}
\langle \boldsymbol{M} \rangle_t &= -\int_T^t \left(e^{\lambda_\tau} \sqrt{-\frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau}} \right)^2 \mathrm{d}\tau, \\
&= \int_T^t e^{2\lambda_\tau} \frac{\mathrm{d}\lambda_\tau}{\mathrm{d}\tau}\, \mathrm{d}\tau, \\
&= \int_{\lambda_T}^{\lambda_t} e^{2\lambda}\, \mathrm{d}\lambda, \\
\varsigma_t &= \frac{1}{2}\left(e^{2\lambda_t} - e^{2\lambda_T} \right),
\end{aligned} \tag{B.18}$$

where we let ς_t denote our new time variable. Now we have a deterministic mapping from the original time to our new time via:

$$\begin{aligned}
\varsigma_t &: [0, T] \to [0, \infty), \\
&t \mapsto \langle \boldsymbol{M} \rangle_t.
\end{aligned} \tag{B.19}$$

Notice that as $t \to 0$ and $\sigma_t \to 0$ we have $\langle \boldsymbol{M} \rangle_t \to \infty$. Such a martingale can be expressed as time-changed Brownian motion, see Theorem B.2, such that $\boldsymbol{M}_t = \boldsymbol{W}_{\langle \boldsymbol{M} \rangle_t}$. For notational simplicity we define the Brownian bridge in this time-changed as

$$\boldsymbol{W}_{\varsigma_t, \varsigma_s} \coloneqq \boldsymbol{W}_{\varsigma_t} - \boldsymbol{W}_{\varsigma_s}. \tag{B.20}$$

Thus we have

$$\boldsymbol{X}_t = \frac{\sigma_t}{\sigma_s} e^{\lambda_s - \lambda_t} \boldsymbol{X}_s + 2\alpha_t \int_{\lambda_s}^{\lambda_t} e^{2(\lambda - \lambda_t)} \boldsymbol{x}_{\lambda_0|\lambda}(\boldsymbol{X}_\lambda)\, \mathrm{d}\lambda + \sqrt{2}\sigma_t e^{-\lambda_t} \boldsymbol{W}_{\varsigma_t, \varsigma_s}, \tag{B.21}$$

thereby finishing the proof.

□

C EXPERIMENTAL DETAILS

All experiments were performed on a single NVIDIA Telsa V100 32GB. We used the Brownian interval (Kidger et al., 2021) from the `torch-sde` package. The faces are from the Face Research Lab London (FRLL) dataset (DeBruine & Jones, 2017). We used the pre-trained LDM model from Rombach et al. (2022) which can be found here: `https://huggingface.co/CompVis/ldm-celebahq-256`.