An optimal control perspective on diffusion-based generative modeling

Anonymous Author(s)
Affiliation
Address
email

Abstract

We establish a connection between stochastic optimal control and generative models based on stochastic differential equations (SDEs) such as recently developed diffusion probabilistic models. In particular, we derive a Hamilton–Jacobi–Bellman equation that governs the evolution of the log-densities of the underlying SDE marginals. This perspective allows to transfer methods from optimal control theory to generative modeling. First, we show that the evidence lower bound is a direct consequence of the well-known verification theorem from control theory. Further, we develop a novel diffusion-based method for sampling from unnormalized densities – a problem frequently occurring in statistics and computational sciences.

1 Introduction

Diffusion (probabilistic) models have established themselves as start-of-the art in generative modeling and likelihood estimation of high-dimensional image data [1, 2, 3]. With methods from different fields, they can be understood from multiple perspectives. In a discrete-time setting, one can, for instance, interpret diffusion models as types of *variational autoencoders* (VAEs) for which the *evidence lower bound* (ELBO) corresponds to a multi-scale denoising score matching objective [1]. In continuous time, the ELBO has been derived as the limit of infinitely deep VAEs [2] or based on an interpretation in terms of stochastic differential equations (SDEs) [4, 5]. The latter encapsulates various methods such as *denoising score matching with Langevin dynamics* (SMLD) and *denoising diffusion probabilistic models* (DDPM) and includes *normalizing flows* as a special case [4, 6].

In this work, we suggest another perspective. We show that the SDE framework naturally connects diffusion models to partial differential equations (PDEs) typically appearing in *stochastic optimal control* and *reinforcement learning*. Using the *Hopf–Cole transformation*, one main insight is that the time-reversed log-density of the diffusion process satisfies a *Hamilton–Jacobi–Bellman* (HJB) equation (Section 2.1). The latter can be connected to a control problem in which one aims to minimize specific control costs with respect to a given controlled dynamics, see [7] and Appendix A.6. We further show that this readily yields the ELBO of the generative model (Section 2.2).

While our main contribution lies in the formal connection between stochastic optimal control and diffusion models, as described in Section 2.3, we moreover demonstrate its practical relevance by transferring methods from control theory to generative modeling. More specifically, in Section 2.4, we design a novel algorithm for sampling from (unnormalized) densities – a problem which frequently occurs in Bayesian statistics, computational physics, chemistry, and biology [8, 9]. As opposed to related approaches [10, 11, 12], our method allows for more flexibility in choosing the initial distribution, offering the possibility to incorporate specific prior knowledge.

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

1.1 Notation

We denote the density of a random variable Z by p_Z. For a stochastic process $Z = (Z_t)_{t\in[0,T]}$ we define the function p_Z by $p_Z(\cdot, t) \coloneqq p_{Z_t}$ for every $t \in [0, T]$. For a time-dependent function f, we denote by $\overleftarrow{f}$ the time-reversal given by $\overleftarrow{f}(t) \coloneqq f(T - t)$. Finally, we define the divergence of matrix-valued functions row-wise. More details on our notation can be found in Appendix A.1.

2 SDE-based generative modeling as an optimal control problem

Diffusion models can naturally be interpreted through the lens of time-continuous stochastic processes [6]. To this end, let us formalize our setting in the general context of SDE-based generative models. We define our model as the stochastic process $X = (X_s)_{s\in[0,T]}$ characterized by the SDE

$$\mathrm{d}X_s = \overleftarrow{\mu}(X_s, s)\,\mathrm{d}s + \overleftarrow{\sigma}(s)\,\mathrm{d}B_s \tag{1}$$

with suitable[1] drift and diffusion coefficients $\mu\colon \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ and $\sigma\colon [0, T] \to \mathbb{R}^{d\times d}$. Learning the model in (1) now corresponds to solving the following problem.

Problem 2.1 (SDE-based generative modeling)**.** *Learn an initial condition X_0 as well as coefficient functions μ and σ such that the distribution of X_T approximates a given data distribution $\mathcal{D}$.*

While, in general, the initial condition X_0 as well as both the coefficient functions μ and σ can be learned, typical applications resort to learning the drift μ only. The following remark justifies that this is sufficient to represent an arbitrary distribution $\mathcal{D}$.

Remark 2.2 (Time-reversed SDE)**.** *Naively, one can achieve $X_T \sim \mathcal{D}$ by setting*

$$X_0 \sim Y_T \quad \textit{and} \quad \mu \coloneqq \sigma\sigma^\top \nabla_x \log(p_Y) - f, \tag{2}$$

where $f\colon \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ and Y is a solution to the SDE

$$\mathrm{d}Y_s = f(Y_s, s)\,\mathrm{d}s + \sigma(s)\,\mathrm{d}B_s, \quad Y_0 \sim \mathcal{D}. \tag{3}$$

This well-known result dates back to [13, 14, 15, 16] which, more generally, state that X can be interpreted as the time-reversal of Y, in the sense that $p_Y = \overleftarrow{p}_X$ almost everywhere, see Appendix A.3. Apparent practical challenges are to sample from $X_0 \sim Y_T$ and to compute the score $\nabla_x \log(p_Y)$. We will see in Section 2.3 how diffusion models provide a working solution.

As already apparent from the optimal drift μ in the previous remark, the reverse-time log-density $\log(\overleftarrow{p}_X)$ of the process X will play a prominent role in deriving a suitable objective for solving Problem 2.1. In the next section, we derive the HJB equation governing the evolution of $\log(\overleftarrow{p}_X)$ and thus providing the bridge to the fields of optimal control and reinforcement learning.

2.1 HJB equation for log-density

We start with the well-known *Fokker-Planck equation* which describes the evolution of a density of solution X to an SDE via the PDE

$$\partial_t p_X = \operatorname{div}_x\left(\operatorname{div}_x\left(\overleftarrow{D} p_X\right) - \overleftarrow{\mu} p_X\right), \tag{4}$$

where we set $D \coloneqq \frac{1}{2}\sigma\sigma^\top$ for notational convenience. This implies that the time-reversed density $\overleftarrow{p}_X$ satisfies a (generalized) *Kolmogorov backwards equation* given by

$$\partial_t \overleftarrow{p}_X = \operatorname{div}_x\left(-\operatorname{div}_x\left(D\overleftarrow{p}_X\right) + \mu\overleftarrow{p}_X\right) = -\operatorname{Tr}\left(D\nabla_x^2 \overleftarrow{p}_X\right) + \mu \cdot \nabla_x \overleftarrow{p}_X + \operatorname{div}_x(\mu)\overleftarrow{p}_X. \tag{5}$$

Note that the second equality follows from the identities for divergences in Appendix A.2 and the fact that σ does not depend on the spatial variable x. Now we use the *Hopf–Cole transformation*, see Appendix A.5, to transform the linear PDE in (5) to a HJB equation prominent in control theory.

Lemma 2.3 (HJB equation for log-density)**.** *Let us define $V \coloneqq -\log(\overleftarrow{p}_X)$. Then V is a solution to the HJB equation*

$$\partial_t V = -\operatorname{Tr}\left(D\nabla_x^2 V\right) + \mu \cdot \nabla_x V - \operatorname{div}_x(\mu) + \frac{1}{2}\left\|\sigma^\top \nabla_x V\right\|^2, \quad V(\cdot, T) = -\log(p_{X_0}). \tag{6}$$

[1] Motivated by Remark 2.2, we start with time-reversed drift and diffusion coefficients $\overleftarrow{\mu}$ and $\overleftarrow{\sigma}$. Further, we assume certain regularity on the coefficient functions of all appearing SDEs, see Appendix A.1

For approaches to directly solve Kolmogorov backwards or HJB equations via deep learning, we refer to, e.g., [11, 17, 18, 19, 20, 21], see also Appendix A.4. Furthermore, the HJB equation allows us to use tools from stochastic control theory to derive a suitable objective for Problem 2.1.

2.2 ELBO derivation using the verification theorem from optimal control

We will derive the ELBO for our generative model (1) using the following fundamental result from control theory, which shows that the solution to a HJB equation such as (6) is related to an optimal control problem, see, [7, 21, 22] and Appendix A.7.

Theorem 2.4 (Verification theorem)**.** *Let V be a solution to the HJB equation in* (6)*. Further, let $\mathcal{U} \subset C^1(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$ be a suitable set of admissible controls and for every control $u \in \mathcal{U}$ let Y^u be the solution to the controlled*[2] *SDE*

$$\mathrm{d}Y_s^u = (\sigma u - \mu)\,(Y_s^u, s)\,\mathrm{d}s + \sigma(s)\,\mathrm{d}B_s. \tag{7}$$

Then it holds almost surely that

$$V(Y_0^u, 0) = \min_{u \in \mathcal{U}} \mathbb{E}\left[\int_0^T \left(\operatorname{div}_x(\mu) + \frac{1}{2}\|u\|^2\right)(Y_s^u, s)\,\mathrm{d}s - \log\left(p_{X_0}(Y_T^u)\right)\middle| Y_0^u\right], \tag{8}$$

where the unique minimum is attained by $u^* := -\sigma^\top \nabla_x V$.

Plugging in the definition of V from Lemma 2.3, this readily yields the following ELBO of our generative model in (1). The corresponding variational gap can be found in Remark A.6.

Corollary 2.5 (Evidence lower bound)**.** *For every $u \in \mathcal{U}$ it holds almost surely that*

$$\log(p_{X_T}(Y_0^u)) \geq \mathbb{E}\left[\int_0^T \left(-\operatorname{div}_x(\mu) - \frac{1}{2}\|u\|^2\right)(Y_s^u, s)\,\mathrm{d}s + \log\left(p_{X_0}(Y_T^u)\right)\middle| Y_0^u\right], \tag{9}$$

where equality is obtained for $u^* := \sigma^\top \nabla_x \log(\overleftarrow{p}_X)$.

Comparing (8) and (9), we see that the ELBO is equivalent to the negative control costs. With the initial condition $Y_0^u \sim \mathcal{D}$, it represents a lower bound on the negative log-likelihood of our generative model. In practice, one can now parametrize u with, for instance, a neural network, and rely on gradient-based optimization to maximize the ELBO using samples from $\mathcal{D}$.

The optimality condition in Corollary 2.5 guarantees that $\overleftarrow{p}_X = p_{Y^{u^*}}$ almost everywhere if we ensure that $X_0 \sim Y_T^{u^*}$, see Appendix A.3. In particular, this implies that $X_T \sim \mathcal{D}$, i.e., our generative model solves Problem 2.1. However, we still face the problem of sampling $X_0 \sim Y_T^{u^*}$ since the distribution of $Y_T^{u^*}$ depends[3] on the initial distribution $\mathcal{D}$. In the next sections we will demonstrate ways to circumvent this problem.

2.3 Connection to denoising score matching objective

This section outlines that, under a reparametrization of the generative model in (1), the ELBO in Corollary 2.5 corresponds to the objective typically used for the training of time-continuous diffusion models. We note that the ELBO in Corollary 2.5 in fact equals the one derived in [4, Theorem 3]. Following the arguments therein and motivated by Remark 2.2 we can now use the reparametrization $\mu := \sigma u - f$ to arrive at an uncontrolled *inference SDE* and a controlled *generative SDE*

$$\mathrm{d}Y_s = f(Y_s, s)\,\mathrm{d}s + \sigma(s)\,\mathrm{d}B_s, \quad Y_0 \sim \mathcal{D}, \quad \text{and} \quad \mathrm{d}X_s^u = \left(\overleftarrow{\sigma}\overleftarrow{u} - \overleftarrow{f}\right)(X_s^u, s)\,\mathrm{d}s + \overleftarrow{\sigma}(s)\,\mathrm{d}B_s. \tag{10}$$

In practice, the coefficients f and σ are usually[4] constructed in such a way that Y is an Ornstein–Uhlenbeck (OU) process with Y_T being approximately distributed according to a standard normal

[2] As usually done, we assume that the initial condition Y_0^u of a solution Y^u to a controlled SDE does not depend on the control u.

[3] In case one has access to samples from the data distribution $\mathcal{D}$, one could use these as initial data Y_0^u in order to simulate $X_0 \sim Y_T^{u^*}$. In doing so, however, one cannot expect to recover the entire distribution $\mathcal{D}$, but only the empirical distribution of the samples.

[4] For coefficients typically used in practice (leading, for instance, to continuous-time analogues of SMLD and DDPM) we refer to [6].

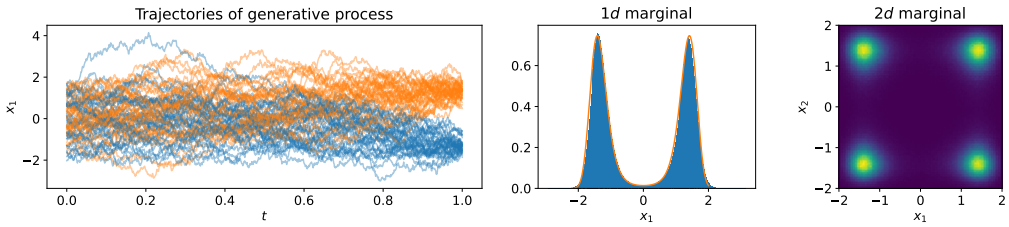

Figure 1: Sampling from $\rho(x) = \exp\left(-\sum_{i=1}^{5}(x_i^2-2)^2 - \frac{1}{2}\sum_{i=6}^{50} x_i^2\right)$ in $d = 50$ using the method described in Section 2.4. Left: First coordinate of the trajectories of X^u in (12) starting from a standard normal and controlled with the learned score u. Middle: Histogram of the first coordinate of the samples X_T^u compared to a reference solution (in orange). Right: $2d$-heatmap of the first two coordinates of X_T^u illustrating that the mode separation is in line with ρ.

distribution. This is why the process Y is said to "diffuse" the data. Setting $X_0^u \sim \mathcal{N}(0, \mathrm{I})$ thus satisfies that $X_0^u \approx Y_T$ in distribution and allows to easily sample X_0^u.

Moreover, under suitable assumptions, the corresponding ELBO can be equivalently rewritten as the denoising score matching objective [23], i.e.,

$$\log(p_{X_T^u}(Y_0)) \geq -\frac{T}{2}\mathbb{E}\left[\left\|u(Y_\tau,\tau) - \sigma^\top(\tau)\nabla\log\left(p_{Y_\tau|Y_0}(Y_\tau|Y_0)\right)\right\|^2 \middle| Y_0\right] + C \tag{11}$$

where $\tau \sim \mathcal{U}([0,T])$, $p_{Y_\tau|Y_0}$ denotes the conditional density of Y_τ given Y_0 (which can be explicitly computed for the OU process), and $C \in \mathbb{R}$ is a constant not depending on u, see Appendix A.8. Due to its simplicity, variants of the objective (11) are typically used in implementations. Note, however, that the setting in this section requires that one has access to samples of $\mathcal{D}$. In the next section, we consider a different scenario, where instead we only have access to the (unnormalized) density of $\mathcal{D}$.

2.4 Sampling from unnormalized densities

In many practical settings, for instance in Bayesian statistics or computational physics, the data distribution $\mathcal{D}$ admits the density $\rho/\mathcal{Z}$, where ρ is known, but computing the normalizing constant $\mathcal{Z} := \int_{\mathbb{R}^d}\rho(x)\,\mathrm{d}x$ is intractable. In this section we propose a novel method based on diffusion models that allows to sample from $\mathcal{D}$. To this end, we interchange the roles of X and Y^u, i.e. consider

$$\mathrm{d}Y_s = \mu(Y_s,s)\,\mathrm{d}s + \sigma(s)\,\mathrm{d}B_s, \quad Y_0 \sim \mathcal{D}, \quad \text{and} \quad \mathrm{d}X_s^u = (\overleftarrow{\sigma}u - \overleftarrow{\mu})(X_s^u,s)\,\mathrm{d}s + \overleftarrow{\sigma}(s)\,\mathrm{d}B_s. \tag{12}$$

Analogously to Theorem 2.4, we then arrive at the control objective

$$-\log\left(\mathcal{Z}p_{Y_T}(X_0^u)\right) = \min_{u\in\mathcal{U}}\mathbb{E}\left[\int_0^T\left(\mathrm{div}_x(\overleftarrow{\mu}) + \frac{1}{2}\|u\|^2\right)(X_s^u,s)\,\mathrm{d}s - \log\left(\rho(X_T^u)\right)\middle| X_0^u\right]. \tag{13}$$

Now, the optimal control $u^* := \overleftarrow{\sigma}^\top\nabla_x\log(\overleftarrow{p}_Y)$ guarantees that $X_T^u \sim \mathcal{D}$ if we ensure that $X_0^u \sim Y_T$, see Appendix A.3. As in the previous chapter, this can approximately be achieved by choosing $X_0^u \sim \mathcal{N}(0,\mathrm{I})$ and μ and σ such that $Y_T \approx X_0^u$. In practice, one can either minimize the control objective (13) using gradient-based optimization or solve the corresponding HJB equation to obtain an approximation to u^*, see Appendix A.9. We note that the former approach is similar to methods independently presented in [11] and [12], where, however, X_0 must follow a Dirac delta distribution.

Our numerical experiments displayed in Figure 1 provide a proof of concept for our method, showing that it allows to sample from high-dimensional multimodal distributions. Further details can be found in Appendix A.10.

3 Conclusion

We believe that the connection of diffusion models to the fields of optimal control and reinforcement learning provides new valuable insights and allows to transfer established tools from one to the respective other field. As first steps, we showed how to easily re-derive the ELBO for time-continuous diffusion models and extended this framework as to sample from unnormalized densities. We leave further experiments and comparisons to future work.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[2] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21696–21707, 2021.

[3] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, pp. 8162–8171, PMLR, 2021.

[4] C.-W. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[5] A. Vahdat, K. Kreis, and J. Kautz, "Score-based generative modeling in latent space," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11287–11302, 2021.

[6] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2020.

[7] H. Pham, *Continuous-time Stochastic Control and Optimization with Financial Applications*. Stochastic Modelling and Applied Probability, Springer Berlin Heidelberg, 2009.

[8] J. S. Liu and J. S. Liu, *Monte Carlo strategies in scientific computing*, vol. 10. Springer, 2001.

[9] G. Stoltz, M. Rousset, *et al.*, *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[10] P. Dai Pra, "A stochastic control approach to reciprocal diffusion processes," *Applied mathematics and Optimization*, vol. 23, no. 1, pp. 313–329, 1991.

[11] L. Richter, *Solving high-dimensional PDEs, approximation of path space measures and importance sampling of diffusions*. PhD thesis, BTU Cottbus-Senftenberg, 2021.

[12] Q. Zhang and Y. Chen, "Path integral sampler: a stochastic control approach for sampling," in *International Conference on Learning Representations*, 2022.

[13] E. Nelson, "Dynamical theories of Brownian motion, princeton univ," *Press, Princeton, NJ*, 1967.

[14] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[15] U. G. Haussmann and E. Pardoux, "Time reversal of diffusions," *The Annals of Probability*, pp. 1188–1205, 1986.

[16] H. Föllmer, "Random fields and diffusion processes," in *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pp. 101–203, Springer, 1988.

[17] L. Richter and J. Berner, "Robust SDE-based variational formulations for solving linear PDEs via deep learning," in *International Conference on Machine Learning*, pp. 18649–18666, PMLR, 2022.

[18] M. Zhou, J. Han, and J. Lu, "Actor-critic method for high dimensional static Hamilton–Jacobi–Bellman partial differential equations based on neural networks," *arXiv preprint arXiv:2102.11379*, 2021.

[19] J. Berner, M. Dablander, and P. Grohs, "Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 16615–16627, 2020.

[20] N. Nüsken and L. Richter, "Interpolating between BSDEs and PINNs–deep learning for elliptic and parabolic boundary value problems," *arXiv preprint arXiv:2112.03749*, 2021.

[21] N. Nüsken and L. Richter, "Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space," *Partial Differential Equations and Applications*, vol. 2, no. 4, pp. 1–48, 2021.

[22] W. H. Fleming and H. M. Soner, *Controlled Markov processes and viscosity solutions*, vol. 25. Springer Science & Business Media, 2006.

[23] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[24] L. Arnold, *Stochastic Differential Equations: Theory and Applications*. A Wiley-Interscience publication, Wiley, 1974.

[25] P. Baldi, *Stochastic Calculus: An Introduction Through Theory and Exercises*. Universitext, Springer International Publishing, 2017.

[26] L. C. Evans, *Partial differential equations*. Providence, R.I.: American Mathematical Society, 2010.

[27] C. Hartmann, L. Richter, C. Schütte, and W. Zhang, "Variational characterization of free energy: Theory and algorithms," *Entropy*, vol. 19, no. 11, p. 626, 2017.

[28] F. Léger and W. Li, "Hopf–Cole transformation via generalized Schrödinger bridge problem," *Journal of Differential Equations*, vol. 274, pp. 788–827, 2021.

[29] W. H. Fleming and R. W. Rishel, *Deterministic and stochastic optimal control*, vol. 1. Springer Science & Business Media, 2012.

[30] R. Van Handel, "Stochastic calculus, filtering, and stochastic control," *Lecture Notes*, 2007.

[31] R. Bellman, *Dynamic programming*. Princeton University Press, 1957.

[32] P.-L. Lions, "Optimal control of diffusion processes and Hamilton-Jacobi–Bellman equations part 2: viscosity solutions and uniqueness," *Communications in partial differential equations*, vol. 8, no. 11, pp. 1229–1276, 1983.

[33] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching.," *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[34] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," in *Uncertainty in Artificial Intelligence*, pp. 574–584, PMLR, 2020.

[35] C. Hartmann and L. Richter, "Nonasymptotic bounds for suboptimal importance sampling," *arXiv preprint arXiv:2102.09606*, 2021.

[36] X. Li, T.-K. L. Wong, R. T. Chen, and D. Duvenaud, "Scalable gradients for stochastic differential equations," in *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882, PMLR, 2020.

[37] P. Kidger, J. Foster, X. C. Li, and T. Lyons, "Efficient and accurate gradients for neural SDEs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18747–18761, 2021.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[39] W. E and B. Yu, "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems," *Communications in Mathematics and Statistics*, vol. 6, no. 1, pp. 1–12, 2018.

A Appendix

A.1 Setting

Let $d, k \in \mathbb{N}$ and $T \in (0, \infty)$. For a random variable X which is absolutely continuous w.r.t. to the Lebesgue measure we write p_X for its density. We denote by B a standard d-dimensional Brownian motion. We say that a continuous $\mathbb{R}^d$-valued stochastic process $Y = (Y_t)_{t\in[0,T]}$ has density $p_Y \colon \mathbb{R}^d \times [0,T] \to [0,\infty)$ if for all $t \in [0,T]$ the random variable Y_t has density $p_Y(\cdot, t)$ w.r.t. to the d-dimensional Lebesgue measure, i.e., for all $t \in [0,T]$ and all measurable $A \subset \mathbb{R}^d$ it holds that

$$\mathbb{P}\left[Y_t \in A\right] = \int_A p_Y(x,t)\,\mathrm{d}x = \int_A p_{Y_t}(x)\,\mathrm{d}x. \tag{14}$$

We denote by $\mathbb{P}_Y$ the law of Y on the space of continuous functions $C([0,T], \mathbb{R}^d)$ equipped with the Borel measure. We assume that the coefficient functions and initial conditions of all appearing SDEs are sufficiently regular, such that their solution processes have densities which can be written as unique solutions to corresponding Fokker-Planck equations, see, for instance, [24, Section 2.6] and [25, Section 10.5] for the details. Further, we write $\overleftarrow{Y}$ for the time-reversed process, satisfying that $\overleftarrow{Y}_t = Y_{T-t}$. For a function $p \colon \mathbb{R}^d \times [0,T] \to \mathbb{R}^k$ we write $\overleftarrow{p}$ for the time-reversed function given by

$$\overleftarrow{p}(x,t) = p(x, T-t), \quad (x,t) \in \mathbb{R}^d \times [0,T]. \tag{15}$$

Finally, we define the divergence of matrix-valued functions row-wise, see Appendix A.2.

A.2 Identities for divergences

Let $A \colon \mathbb{R}^d \to \mathbb{R}^{d\times d}$, $v \colon \mathbb{R}^d \to \mathbb{R}^d$, and $s \colon \mathbb{R}^d \to \mathbb{R}$. For a vector-valued function, we denote by ∇v its Jacobian, and, for matrix-valued functions, we denote by

$$\mathrm{Tr}(A) \coloneqq \sum_{i=1}^d A_{ii} \tag{16}$$

the trace of its output and we define the divergence row-wise, i.e.,

$$\mathrm{div}(A) \coloneqq (\mathrm{div}(A_{i\cdot}))_{i=1}^d = \sum_{j=1}^d \partial_{x_j} A_{\cdot j}, \tag{17}$$

where $A_{i\cdot}$ and $A_{\cdot j}$ denote the i-th row and j-th column, respectively. Then, the following identities hold:

1. $\mathrm{div}(\mathrm{div}(A)) = \sum_{i,j=1}^d \partial_{x_i}\partial_{x_j} A_{ij}$
2. $\mathrm{div}(vs) = \mathrm{div}(v)s + v \cdot \nabla s$
3. $\mathrm{div}(As) = \mathrm{div}(A)s + A\nabla s$
4. $\mathrm{div}(Av) = \mathrm{div}(A^\top) \cdot v + \mathrm{Tr}(A\nabla v)$

A.3 Time-reversed SDEs

The next theorem shows that the marginals of a time-reversed Itô process can be represented as marginals of another Itô process, see [4, 6, 13, 14, 15, 16]. We present a formulation from [4, Appendix G] which derives a whole family of processes (parametrized by a function λ). The relations stated in equation (2) follow for the choice $\lambda = 0$.

Theorem A.1 (Reverse-time SDE)**.** *Let $Y = (Y_s)_{s\in[0,T]}$ be the solution to the SDE*

$$\mathrm{d}Y_s = f(Y_s, s)\,\mathrm{d}s + \sigma(Y_s, s)\,\mathrm{d}B_s \tag{18}$$

and assume that Y has density p_Y, which satisfies the Fokker-Planck equation given by

$$\partial_t p_Y = \mathrm{div}_x\left(\mathrm{div}_x\left(Dp_Y\right) - \mu\, p_Y\right), \tag{19}$$

where $D \coloneqq \frac{1}{2}\sigma\sigma^\top$. *For every* $\lambda \in C^2([0,T],[0,1])$ *the solution* $X = (X_s)_{s\in[0,T]}$ *to the reverse SDE*

$$\mathrm{d}X_s = \overleftarrow{\mu}^{(\lambda)}(X_s, s)\,\mathrm{d}s + \overleftarrow{\sigma}^{(\lambda)}(X_s, s)\,\mathrm{d}B_s, \quad X_0 \sim Y_T, \tag{20}$$

with

$$\mu^{(\lambda)} \coloneqq (2-\lambda)\,\mathrm{div}_x\left(D\right) + (2-\lambda)D\nabla_x \log(p_Y) - \mu, \tag{21}$$

and

$$\sigma^{(\lambda)} \coloneqq \sqrt{1-\lambda}\sigma \tag{22}$$

has density p_X *given by*

$$p_X(\cdot, t) = \overleftarrow{p}_Y(\cdot, t) \tag{23}$$

almost everywhere for every $t \in [0,T]$. *In other words, for every* $t \in [0,T]$ *it holds that* $Y_{T-t} \sim X_t$.

Proof. Using the Fokker-Planck equation in (19), we observe that

$$\partial_t \overleftarrow{p}_Y = \mathrm{div}_x\left(-\,\mathrm{div}_x\left(\overleftarrow{D}\overleftarrow{p}_Y\right) + \overleftarrow{\mu}\overleftarrow{p}_Y\right). \tag{24}$$

The negative divergence, originating from the chain-rule, prohibits us from directly viewing the above equation as a Fokker-Planck equation. We can, however, use the identities in Appendix A.2, to show that

$$\mathrm{div}_x\left(Dp_Y\right) = \mathrm{div}_x\left(D\right)p_Y + D\nabla_x p_Y = \left(\mathrm{div}_x\left(D\right) + \nabla_x \log(p_Y)\right)p_Y. \tag{25}$$

This implies that we can rewrite (24) as

$$\partial_t \overleftarrow{p}_Y = \mathrm{div}_x\left((1-\lambda)\,\mathrm{div}_x\left(\overleftarrow{D}\overleftarrow{p}_Y\right) - (2-\lambda)\,\mathrm{div}_x\left(\overleftarrow{D}\overleftarrow{p}_Y\right) + \overleftarrow{\mu}\overleftarrow{p}_Y\right) \tag{26a}$$

$$= \mathrm{div}_x\left(\mathrm{div}_x\left(\overleftarrow{D}^{(\lambda)}\overleftarrow{p}_Y\right) - \overleftarrow{\mu}^{(\lambda)}\overleftarrow{p}_Y\right), \tag{26b}$$

where $D^{(\lambda)} \coloneqq \frac{1}{2}\sigma^{(\lambda)}(\sigma^{(\lambda)})^\top$. As the PDE in (26b) defines a valid Fokker-Planck equation associated to the reverse SDE given by 20, this proves the claim. □

A.4 Further details on the HJB equation

In order to solve Problem 2.1, one might be tempted to rely on classical methods to approximate the solution of the HJB equation from Lemma 2.3 directly. However, in the setting of Remark 2.2, one should note that the optimal drift,

$$\mu = \sigma\sigma^\top \nabla_x \log(p_Y) - f = -\sigma\sigma^\top \nabla_x V - f, \tag{27}$$

contains the solution itself. Plugging it in to (6) we get the equation

$$\partial_t V = \mathrm{Tr}\left(D\nabla_x^2 V\right) - f\cdot\nabla_x V + \mathrm{div}(f) - \frac{1}{2}\|\sigma^\top\nabla_x V\|^2, \quad V(\cdot, T) = -\log\left(p_{X_0}\right). \tag{28}$$

Likewise, when applying the Hopf–Cole transformation from Appendix A.5 to p_Y directly, i.e. considering $V = -\log\left(p_Y\right)$, where Y is the solution to SDE (3), we get the same PDE. We note, however, that the signs in (28) do not match with typical HJB equations from control theory. We therefore need to consider the time-reversed function $\overleftarrow{V}$, which brings the HJB equation

$$\partial_t \overleftarrow{V} = -\,\mathrm{Tr}\left(\overleftarrow{D}\nabla_x^2\overleftarrow{V}\right) + \overleftarrow{f}\cdot\nabla_x\overleftarrow{V} - \mathrm{div}(\overleftarrow{f}) + \frac{1}{2}\|\overleftarrow{\sigma}^\top\nabla_x\overleftarrow{V}\|^2, \quad \overleftarrow{V}(\cdot, T) = -\log\left(p_{X_T}\right). \tag{29}$$

Unfortunately, for many applications the terminal condition is not available since p_{X_T} is not known, which seems to render methods for numerically solving the HJB equation useless. In the case of sampling from (unnormalized) densities, however, the situation is different, see Section 2.4.

A.5 Hopf–Cole transformation

The following lemma details the relation of the HJB equation in (6) and the linear Kolmogorov backwards equation[5] in (5). A proof can, e.g., be found in [26, Section 4.4.1] and [17, Appendix G]. Lemma 2.3 follows with the choices $b \coloneqq -\mu$, $h \coloneqq \mathrm{div}_x(\mu)$, and $g \coloneqq -\log(p_{X_0})$.

[5] For $\mathrm{div}_x(\mu) = 0$ this can be viewed as the adjoint of the Fokker-Planck equation.

Lemma A.2 (Hopf–Cole transformation)**.** *Let* $h\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}$ *and let* $p \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ *solve the linear PDE*

$$\partial_t p = -\frac{1}{2}\operatorname{Tr}(\sigma\sigma^\top \nabla_x^2 p) - b \cdot \nabla_x p + hp. \tag{30}$$

Then $V := -\log(p)$ *satisfies the HJB equation*

$$\partial_t V = -\frac{1}{2}\operatorname{Tr}(\sigma\sigma^\top \nabla_x^2 V) - b \cdot \nabla_x V - h + \frac{1}{2}\left\|\sigma^\top \nabla_x V\right\|^2. \tag{31}$$

For further applications of the Hopf–Cole transformation we refer, for instance, to [22, 27, 28].

A.6 Brief introduction to stochastic optimal control

In this section we shall provide a brief introduction to stochastic optimal control. For details and further reading we refer the interested reader to the monographs [7, 22, 29, 30]. Loosely speaking, stochastic control theory deals with identifying optimal strategies in noisy environments, in our case time-continuous stochastic processes defined by the SDE

$$\mathrm{d}X_s^U = \widetilde{\mu}(X_s^U, s, U_s)\,\mathrm{d}s + \widetilde{\sigma}(X_s^U, s, U_s)\,\mathrm{d}B_s, \tag{32}$$

where $\widetilde{\mu}$ and $\widetilde{\sigma}$ are suitable functions, B is a d-dimensional Brownian motion, and U is a progressively measurable, $\mathbb{R}^d$-valued random control process. For ease of presentation, we focus on the frequent case where U is a Markov control, which means that there exists a deterministic function $u \in \mathcal{U} \subset C(\mathbb{R}^d \times [0,T], \mathbb{R}^d)$, such that $U_s = u(X_s^U, s)$. In other words, the randomness of the process U is only coming from the stochastic process X^U. The function class $\mathcal{U}$ then defines the set of admissible controls. Very often one considers the special cases

$$\widetilde{\mu}(x,s,u) = \mu(x,s) + \sigma(s)u(x,s) \qquad \text{and} \qquad \widetilde{\sigma}(x,s,u) = \sigma(s), \tag{33}$$

where μ and σ might correspond to the choices taken in Section 2.3 and one might think of u as a steering force as to reach a certain target.

The goal is now to minimize specified control costs with respect to the control u. To this end, we can define the cost functional

$$J(u; x_{\text{init}}, 0) = \mathbb{E}\left[\int_0^T \widetilde{h}(X_s^U, s, u(X_s^U, s))\mathrm{d}s + g(X_T^U)\middle| X_0^U = x_{\text{init}}\right], \tag{34}$$

where $\widetilde{h} : \mathbb{R}^d \times [0,T] \times \mathbb{R}^d \to \mathbb{R}$ specifies running and $g : \mathbb{R}^d \to \mathbb{R}$ represents terminal costs. Furthermore we can define the *cost-to-go* as

$$J(u; x, t) = \mathbb{E}\left[\int_t^T \widetilde{h}(X_s^U, s, u(X_s^U, s))\mathrm{d}s + g(X_T^U)\middle| X_t^U = x\right], \tag{35}$$

now depending on respective initial values $(x,t) \in \mathbb{R}^d \times [0,T]$. The objective in optimal control is now to minimize this quantity over all admissible control functions $u \in \mathcal{U}$ and we therefore introduce the so-called *value function*

$$V(x,t) = \inf_{u \in \mathcal{U}} J(u; x, t) \tag{36}$$

as the optimal costs conditioned on being in position x at time t.

Motivated by the *dynamic programming principle* [31], one can then derive the main result from control theory, namely that the function V defined in (36) fulfills a nonlinear PDE, which can thus be interpreted as the determining equation for optimality[6].

[6] In practice, solutions to optimal control problems may not posses enough regularity in order to formally fulfill the HJB equation, such that a complete theory of optimal control needs to introduce an appropriate concept of weak solutions, leading to so-called viscosity solutions that have been extensively studied for instance in [22, 32].

Theorem A.3 (Verification theorem for general HJB equation)**.** *Let $V \in C^{2,1}(\mathbb{R}^d \times [0,T], \mathbb{R})$ fulfill the PDE*

$$\partial_t V = -\inf_{\alpha \in \mathbb{R}^d} \left\{ \widetilde{h}(\cdot,\cdot,\alpha) + \widetilde{b}(\cdot,\cdot,\alpha) \cdot \nabla_x V + \frac{1}{2} \operatorname{Tr}\left((\widetilde{\sigma}\widetilde{\sigma}^\top)(\cdot,\cdot,\alpha) \nabla_x^2 V \right) \right\}, \quad V(\cdot,T) = g, \tag{37}$$

such that $\sup_{(x,t)\in\mathbb{R}^d\times[0,T]} \frac{\|V(x,t)\|}{1+\|x\|^2} < \infty$, and suppose there exists a measurable function $\mathcal{U} \ni u^* : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ that attains the above infimum for all $(x,t) \in \mathbb{R}^d \times [0,T]$. Further, let the correspondingly controlled solution X^{U^*} to the SDE in (32) *with $U_s^* := u^*(X_s^{U^*}, s)$ have a strong solution. Then V coincides with the value function as defined in* (36) *and u^* is an optimal Markovian control.*

Let us appreciate the fact that the infimum in the HJB equation in (37) is merely over the set $\mathbb{R}^d$ and not over the function space $\mathcal{U}$ as in (36), so the minimization reduces to a pointwise operation. A proof of Theorem A.3 can for instance be found in [7, Theorem 3.5.2].

In many applications, in addition to the choices (33), one considers the special form of running costs

$$\widetilde{h}(x,s,u(x,s)) = h(x,s) + \frac{1}{2}\|u(x,s)\|^2, \tag{38}$$

where $h : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$. In this setting the minimization appearing in the general HJB equation (37) can be solved explicitly, therefore leading to a closed-form PDE, as made precise with the following Corollary.

Corollary A.4 (HJB equation with quadratic running costs)**.** *If the diffusion coefficient $\widetilde{\sigma}$ does not depend on the control, the control enters additively in the drift as in* (33)*, and the running costs take the form*

$$\widetilde{h}(x,s,u(x,s)) = h(x,s) + \frac{1}{2}\|u(x,s)\|^2, \tag{39}$$

then the general HJB equation (A.3) *can be stated in closed form as the HJB equation* (31)*.*

Proof. We formally compute

$$\inf_{\alpha\in\mathbb{R}^d} \left\{ \widetilde{h}(\cdot,\cdot,\alpha) + \widetilde{\mu}(\cdot,\cdot,\alpha) \cdot \nabla_x V \right\} = h + \mu \cdot \nabla_x V + \inf_{\alpha\in\mathbb{R}^d} \left\{ \frac{1}{2}\|\alpha\|^2 + \sigma\alpha \cdot \nabla_x V \right\}, \tag{40}$$

and realize that the infimum is attained when choosing $\alpha^* = -\sigma^\top \nabla_x V(x,t)$ for each corresponding $(x,t) \in \mathbb{R}^d \times [0,T]$, resulting in the optimal control function $u^* = -\sigma^\top \nabla_x V$. Plugging this into the general HJB equation (37), we readily get the PDE in (31). □

A.7 Verification theorem

The verification theorem is a classical result in optimal control, see, for instance, [21, Theorem 2.2], [22, Theorem IV.4.4], [7, Theorem 3.5.2], and Appendix A.6. For the interested reader, we provide the theorem and a self-contained proof using Itô's lemma in the following. Theorem 2.4 follows with the choices $\tau := 0$, $b := -\mu$, $h := \operatorname{div}_x(\mu)$, and $g := -\log(p_{X_0})$.

Theorem A.5 (Verification theorem)**.** *Let V be a solution to the HJB equation in* (31)*. Further, let $t \in [0,T]$ and define the set of admissible controls by*

$$\mathcal{U} := \left\{ u \in C^1(\mathbb{R}^d \times [\tau,T], \mathbb{R}^d) \colon \sup_{(x,t)\in\mathbb{R}^d\times[\tau,T]} \frac{\|u(x,t)\|}{1+\|x\|} < \infty \right\}. \tag{41}$$

For every control $u \in \mathcal{U}$ let $Z^u = (Z_t^u)_{t\in[\tau,T]}$ be the solution to the controlled SDE

$$\mathrm{d}Z_s^u = (\sigma u + b)(Z_s^u, s)\,\mathrm{d}s + \sigma(s)\,\mathrm{d}B_s \tag{42}$$

and let the cost of the control u be defined by

$$J(u) := \mathbb{E}\left[\int_t^T \left(h + \frac{1}{2}\|u\|^2 \right)(Z_s^u, s)\,\mathrm{d}s + g(Z_T^u) \middle| Z_t^u \right]. \tag{43}$$

Then for every $u \in \mathcal{U}$ it holds almost surely that

$$V(Z_t^u, t) + \mathbb{E}\left[\frac{1}{2}\int_t^T \left\|\sigma^\top \nabla_x V + u\right\|^2 (Z_s^u, s)\,\mathrm{d}s \middle| Z_t^u\right] = J(u). \tag{44}$$

In particular, this implies that $V(Z_t^u, t) = \min_{u\in\mathcal{U}} J(u)$ almost surely, where the unique minimum is attained by $u^* := -\sigma^\top \nabla_x V$.

Proof. Let us derive the verification theorem directly from Itô's lemma, which, under suitable assumptions, states that

$$V(Z_T^u, T) - V(Z_t^u, t) = \int_t^T \left(\partial_t V + (\sigma u + b)\cdot \nabla_x V + \mathrm{Tr}\left(D\nabla_x^2 V\right)\right)(Z_s^u, s)\,\mathrm{d}s + S \tag{45}$$

almost surely, where

$$S := \int_t^T (\sigma^\top \nabla_x V)(Z_s^u, s)\cdot\,\mathrm{d}B_s, \tag{46}$$

see, e.g., [25, Theorem 8.3]. Combining this with the fact that V solves the HJB equation in (31) and the simple calculation

$$\frac{1}{2}\|\sigma^\top \nabla_x V + u\|^2 = \frac{1}{2}\left(\sigma^\top \nabla_x V + u\right)\cdot\left(\sigma^\top \nabla_x V + u\right) \tag{47a}$$

$$= \frac{1}{2}\|\sigma^\top \nabla_x V\|^2 + \left(\sigma^\top \nabla_x V\right)\cdot u + \frac{1}{2}\|u\|^2, \tag{47b}$$

shows that

$$V(Z_t^u, t) = \int_t^T \left(h + \frac{1}{2}\|u\|^2 - \frac{1}{2}\left\|\sigma^\top \nabla_x V + u\right\|^2\right)(Z_s^u, s)\,\mathrm{d}s + g(Z_T^u) - S \tag{48}$$

almost surely. Under mild regularity assumptions, the stochastic integral S has zero expectation conditioned on Z_t^u, which proofs the claim. □

Remark A.6 (Variational gap)**.** *We can interpret the term*

$$\mathbb{E}\left[\frac{1}{2}\int_t^T \left\|\sigma^\top \nabla_x V + u\right\|^2 (Z_s^u, s)\,\mathrm{d}s \middle| Z_t^u\right] \tag{49}$$

in (44) *as variational gap specifying the misfit of the current and the optimal control objective. In the setting of Corollary 2.5 it takes the form*

$$\mathbb{E}\left[\frac{1}{2}\int_0^T \left\|u - \sigma^\top \nabla_x \log(\overleftarrow{p}_X)\right\|^2 (Y_s^u, s)\,\mathrm{d}s \middle| Y_0\right] \tag{50}$$

and can be compared to [4, Theorem 4], where, however, the factor $1/2$ seems to be missing.

Remark A.7 (Path measure interpretation)**.** *In [21, Proposition 3.5] it is shown that the ELBO in Corollary 2.5 can be rewritten as the difference of the evidence $\log(p_{X_T}(Y_0))$ and the Kullback-Leibler divergence $D_{\mathrm{KL}}(\mathbb{P}_{Y^u}|\mathbb{P}_{Y^{u^*}})$, where we denote by $\mathbb{P}_{Y^u}$ the path measure of Y^u, i.e., the law of Y on the space of continuous functions $C([0,T],\mathbb{R}^d)$ equipped with the Borel measure.*

A.8 ELBO formulations

Here we provide details on the connection of the ELBO to the denoising score matching objective. Using the reparametrization in (10), i.e. taking $\mu := \sigma u - f$, Corollary 2.5 yields that

$$\log(p_{X_T^u}(Y_0)) \geq \mathbb{E}\left[\int_0^T \left(-\mathrm{div}_x(\sigma u - f) - \frac{1}{2}\|u\|^2\right)(Y_s, s)\,\mathrm{d}s + \log\left(p_{X_0^u}(Y_T)\right) \middle| Y_0\right]. \tag{51}$$

The next lemma shows that the ELBO in (51) equals the denoising score matching objective [23] up to a constant, which does not depend on the control u.

Lemma A.8 (Connection to denoising score matching)**.** *It holds that*

$$\mathbb{E}\left[\int_0^T \left(-\operatorname{div}_x(\sigma u - f) - \frac{1}{2}\|u\|^2\right)(Y_s, s)\,\mathrm{d}s + \log\left(p_{X_0^u}(Y_T)\right)\middle| Y_0\right] \\ = -\frac{T}{2}\mathbb{E}\left[\left\|u(Y_\tau,\tau) - \sigma^\top(\tau)\nabla\log\left(p_{Y_\tau|Y_0}(Y_\tau|Y_0)\right)\right\|^2 \middle| Y_0\right] + C, \tag{52}$$

where $\tau \sim \mathcal{U}([0,T])$*,* $p_{Y_\tau|Y_0}$ *denotes the conditional density of* Y_τ *given* Y_0*, and*

$$C \coloneqq \mathbb{E}\Big[\log(p_{X_0^u})(Y_T) + T\operatorname{div}_x(f)(Y_\tau,\tau) + \frac{T}{2}\left\|\sigma^\top(\tau)\nabla\log\left(p_{Y_\tau|Y_0}(Y_\tau|Y_0)\right)\right\|^2 \Big| Y_0\Big]. \tag{53}$$

Proof. The proof closely follows the one in [4, Appendix A]. For notational convenience, let us define $p(x,s) \coloneqq p_{Y_s|Y_0}(x|Y_0)$ for every $x \in \mathbb{R}^d$ and $s \in [0,T]$. Note that we have

$$\frac{1}{2}\left\|u - \sigma^\top\nabla_x\log(p)\right\|^2 = \frac{1}{2}\|u\|^2 - u\cdot\left(\sigma^\top\nabla\log(p)\right) + \frac{1}{2}\left\|\sigma^\top\nabla\log(p)\right\|^2. \tag{54}$$

Using Fubini's theorem and a Monte-Carlo approximation, this implies that, under mild regularity conditions, the quantity in (52) can be written as

$$\mathbb{E}\left[\int_0^T \left(u\cdot\left(\sigma^\top\nabla_x\log(p)\right) + \operatorname{div}_x(f) - \frac{1}{2}\|u\|^2\right)(Y_s,s)\,\mathrm{d}s + \log\left(p_{X_0^u}(Y_T)\right)\middle| Y_0\right]. \tag{55}$$

Now, we focus on the term $u\cdot\left(\sigma^\top\nabla_x\log(p)\right)(Y_s,s)$ for fixed $s\in[0,T]$. Using the identities for divergences in Appendix A.2, one can show that

$$\operatorname{div}_x(\sigma u p) - \operatorname{div}_x(\sigma u)p = (\sigma u)\cdot\nabla_x p = (\sigma u)\cdot\nabla_x\log(p)\,p = u\cdot\left(\sigma^\top\nabla_x\log(p)\right)p. \tag{56}$$

Further, Stokes' theorem guarantees that under suitable assumptions it holds that

$$\int_{\mathbb{R}^d}\operatorname{div}_x(\sigma u p)(x,s)\,\mathrm{d}x = 0. \tag{57}$$

Thus, using (56) and (57), we have that

$$\mathbb{E}\left[-\operatorname{div}_x(\sigma u)(Y_s,s)|Y_0\right] = -\int_{\mathbb{R}^d}\operatorname{div}_x(\sigma u)(x,s)p(x,s)\,\mathrm{d}x \tag{58a}$$

$$= \int_{\mathbb{R}^d} u(x,s)\cdot\left(\sigma^\top\nabla_x\log(p)\right)(x,s)p(x,s)\mathrm{d}x \tag{58b}$$

$$= \mathbb{E}\left[u(Y_s,s)\cdot\left(\sigma^\top\nabla_x\log(p)\right)(Y_s,s)\middle|Y_0\right]. \tag{58c}$$

Combining this with (55) finishes the proof. □

Note that one can also establish equivalences to explicit, implicit, and sliced score matching [33, 34], see [4, Appendix A]. Using the interpretation of the ELBO in terms of Kullback-Leibler divergences, see [4, Theorem 5] and also Remark A.7, one can further derive the ELBO for discrete-time diffusion models as presented in [1, 2].

A.9 Sampling from densities

In the setting of Section 2.4, we now present a method different from minimizing the objective (13) in order to obtain an approximation to u^*. As the Kolmogorov backwards equation is linear, the scaled density $p \coloneqq \mathcal{Z}\overleftarrow{p}_Y$ also satisfies a Kolmogorov backwards equation given by

$$\partial_t p = -\operatorname{Tr}\left(\overleftarrow{D}\nabla_x^2 p\right) + \overleftarrow{\mu}\cdot\nabla_x p + \operatorname{div}_x(\overleftarrow{\mu})p, \quad \overleftarrow{p}_Y(\cdot,T) = \rho. \tag{59}$$

Deriving the HJB equation using the Hopf–Cole transform $V \coloneqq \log\left(\mathcal{Z}\overleftarrow{p}_Y\right)$ from Appendix A.5 yields that

$$\partial_t V = -\operatorname{Tr}(\overleftarrow{D}\nabla_x^2 V) + \overleftarrow{\mu}\cdot\nabla_x V - \operatorname{div}_x(\overleftarrow{\mu}) + \frac{1}{2}\left\|\overleftarrow{\sigma}^\top\nabla_x V\right\|^2, \quad \overleftarrow{p}_Y(\cdot,T) = \log(\rho). \tag{60}$$

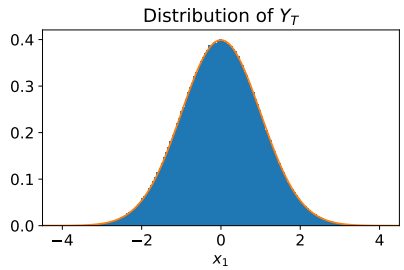

Figure 2: Histogram of the first component of Y_T (in blue) when starting the corresponding process Y in $Y_0 \sim \mathcal{D}$, where $\mathcal{D}$ corresponds to ρ in (63), compared to a standard Gaussian density (in orange).

As ρ is known, viable strategies to learn u^* would be to solve the PDEs for p or V via deep learning, see, e.g. [11, 17, 18, 19, 20, 21], and then use automatic differentiation to compute

$$u^* := \overleftarrow{\sigma}^\top \nabla_x \log\left(\overleftarrow{p}_Y\right) = \overleftarrow{\sigma}^\top \nabla_x \log\left(p\right) = \overleftarrow{\sigma}^\top \nabla_x V \tag{61}$$

as the score is invariant to rescaling. This adds another method to directly optimizing the control costs in (13) for approximately learning $u \approx u^*$.

Using such u, one simulates X^u as defined in (12), e.g., using the *Euler–Maruyama scheme*, such that X_T^u represent approximate samples from $\mathcal{D}$. As done in [12], one can employ *importance sampling* to correct for suboptimal approximations $u \neq u^*$, however, potentially at the cost of an increased variance of corresponding estimated quantities of interests, see [35]. Further, one can use the *stochastic adjoint sensitivity method* to efficiently compute the gradients of the ELBO w.r.t. the parameters of the control u using high-order adaptive solvers, see also [36, 37].

A.10 Implementation details

For our experiments we consider the following setup. We set $T = 1$, $f = -2x$, and $\sigma = 2\,\mathrm{Id}$ such that the inference SDE in (10) corresponds to an Ornstein-Uhlenbeck process for which we have

$$Y_T | Y_0 \sim \mathcal{N}\left(e^{-2} Y_0, 1 - e^{-4}\right). \tag{62}$$

We can numerically check whether (the unconditional) Y_T is indeed close to a standard normal distribution, as demanded by Algorithm 1. To this end, we consider samples from $\mathcal{D}$ (in this case corresponding to the example in Figure 1) and let the process Y run according to the inference SDE in (3). We can now compare the empirical distributions of the different components, which each need to follow a one-dimensional Gaussian. Figure 2 shows that this is indeed the case, where we take the first component as an example.

We approximate the control u with a *DenseNet* [38, 39], which is optimized using the Adam optimizer with batch size $K = 50$. We approximate the expectation in (13) with 10 samples and discretize the SDE X^u initially with time-step $\Delta t = 0.01$ and later with $\Delta t = 0.001$. The optimization routine is summarized in Algorithm 1.

In our numerical example in Section 2.4 we consider the unnormalized density

$$\rho(x) = \exp\left(-\sum_{i=1}^{5}(x_i^2 - 2)^2 - \frac{1}{2}\sum_{i=6}^{50} x_i^2\right). \tag{63}$$

Note that, due to the double well structure of the negative log-density (also know as *potential*), the density has $2^5 = 32$ separated modes. In Figure 3 we repeat the same experiment, however, now in $d = 20$ with

$$\rho(x) = \exp\left(-\sum_{i=1}^{2}(x_i^2 - 2)^2 - \frac{1}{2}\sum_{i=3}^{20} x_i^2\right), \tag{64}$$

which consist of only $2^2 = 4$ modes.

In applications one is often interested in computing expected values of the form

$$\Xi := \mathbb{E}\left[\xi(Y_0)\right], \tag{65}$$

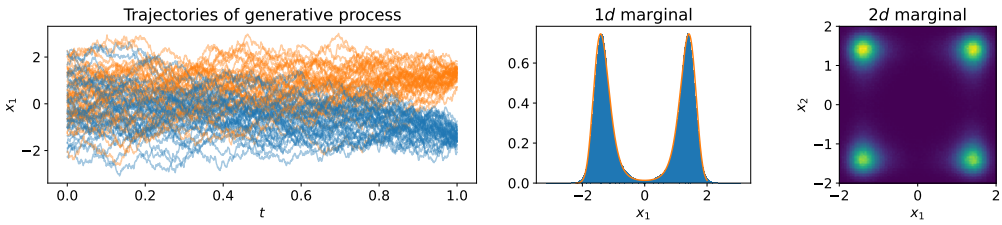

Figure 3: Sampling from $\rho(x) = \exp\left(-\sum_{i=1}^{2}(x_i^2-2)^2 - \frac{1}{2}\sum_{i=3}^{20} x_i^2\right)$ in $d = 20$ using the method described in Section 2.4. Left: First coordinate of the trajectories of X^u in (12) starting from a standard normal and controlled with the learned score u. Middle: Histogram of the first coordinate of the samples X_T^u compared to a reference solution (in orange). Right: $2d$-heatmap of the first two coordinates of X_T^u illustrating that the mode separation is in line with ρ.

Algorithm 1 Training the control in Section 2.4 via deep learning.

Input: Neural network Φ with initial parameters $\theta^{(0)}$, optimizer method step for updating the parameters, number of steps M, batch size K

Output: parameters $\theta^{(M)}$

for $m \leftarrow 0, \dots, M-1$ **do**

  $(x^{(k)})_{k=1}^K \leftarrow$ sample from $\mathcal{N}(0, \mathrm{I})^{\otimes K}$

  $J(\theta) \leftarrow$ Compute the average cost in (13) with $X_0^u = x^{(k)},\ k = 1, \dots, K,$ and $u = \Phi(\theta^{(m)})$

  $\theta^{(m+1)} \leftarrow \mathrm{step}\left(\theta^{(m)}, \nabla J(\theta)\right)$

end for

where $Y_0 \sim \mathcal{D}$ follows some distribution and $\xi : \mathbb{R}^d \to \mathbb{R}$ specifies an observable of interest. For an example, let us consider $\xi(x) = \|x\|^2$ and approximate (65) by creating samples via Algorithm 1, resulting in the random variable $X_T^u \approx Y_0$ (where the approximation is meant in distribution). We can then approximate Ξ in (65) via Monte Carlo sampling by

$$\widehat{\Xi} := \frac{1}{K}\sum_{k=1}^{K} \xi(X_T^{u,(k)}), \tag{66}$$

where $K \in \mathbb{N}$ specifies the sample size. In our multimodal examples we can compute a reference solution of (65), which we shall denote by Ξ_{ref}, by numerical integration since ρ factorizes in the dimensions. We can now compute the relative error $r_\Xi := |(\widehat{\Xi} - \Xi_{\mathrm{ref}})/\Xi_{\mathrm{ref}}|$, which can be viewed as an evaluation of the sample quality on a global level. For the approximation of our unnormalized density ρ in $d = 50$, which is displayed in Figure 1, we get a relative error of $r_\Xi = 2.78 \cdot 10^{-04}$ when using $K = 10^6$ samples. For the example in $d = 20$, which corresponds to Figure 3, the relative error is $r_\Xi = 4.76 \cdot 10^{-04}$.