4th Symposium on Advances in Approximate Bayesian Inference, 2021 1–22

Bayesian Learning via Neural Schrödinger-Föllmer Flows

Francisco Vargas FAV25@CAM.AC.UK
Andrius Ovsianas AO464@CAM.AC.UK
David Fernandes DLF28@BATH.AC.UK
Mark Girolami MAG92@CAM.AC.UK
Neil Lawrence NDL21@CAM.AC.UK
Nikolas Nüsken NUESKEN@UNI-POTSDAM.DE

Abstract

In this work we explore a new framework for approximate Bayesian inference in large datasets based on stochastic control. We advocate stochastic control as a finite time alternative to popular steady-state methods such as stochastic gradient Langevin dynamics (SGLD). Furthermore, we discuss and adapt the existing theoretical guarantees of this framework and establish connections to already existing VI routines in SDE-based models.[1]

1. Introduction

Steering a stochastic flow from one distribution to another across the space of probability measures is a well-studied problem initially proposed in Schrödinger (1932). There has been recent interest in the machine learning community in these methods for generative modelling, sampling, dataset imputation and optimal transport (Wang et al., 2021; De Bortoli et al., 2021; Huang et al., 2021; Bernton et al., 2019; Vargas et al., 2021; Chizat et al., 2020; Cuturi, 2013; Maoutsa and Opper, 2021; Reich, 2019).

We consider a particular instance of the Schrödinger bridge problem (SBP), known as the Schrödinger-Föllmer process (SFP). In machine learning, this process has been proposed for sampling and generative modelling (Huang et al., 2021; Tzen and Raginsky, 2019b) and in molecular dynamics for rare event simulation and importance sampling (Hartmann and Schütte, 2012; Hartmann et al., 2017); here we apply it to Bayesian inference. We show that a control-based formulation of the SFP has deep-rooted connections to variational inference and is particularly well suited to Bayesian inference in high dimensions. This capability arises from the SFP's characterisation as an optimisation problem and its parametrisation through neural networks (Tzen and Raginsky, 2019b). Finally, due to the variational characterisation that these methods possess, we believe that many low-variance estimators (Richter et al., 2020; Nüsken and Richter, 2021; Roeder et al., 2017) are applicable to the SFP formulation we consider.

We reformulate the Bayesian inference problem by constructing a stochastic process $\boldsymbol{\Theta}_t$ which at a fixed time $t = 1$ will generate samples from a pre-specified posterior $p(\boldsymbol{\theta}|\boldsymbol{X})$ with dataset $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^N$ (i.e. $\mathrm{Law}\boldsymbol{\Theta}_1 = p(\boldsymbol{\theta}|\boldsymbol{X})$), where the model is given by:

$$\mathbf{x}_i|\boldsymbol{\theta} \sim p(\boldsymbol{x}_i|\boldsymbol{\theta}), \quad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \tag{1}$$

1. For a longer version of the paper please visit https://arxiv.org/pdf/2111.10510.pdf

© F. Vargas, A. Ovsianas, D. Fernandes, M. Girolami, N. Lawrence & N. Nüsken.

and the prior $p(\boldsymbol{\theta})$ and the likelihood $p(\mathbf{x}_i|\boldsymbol{\theta})$ are user-specified. Our target is $\pi_1(\boldsymbol{\theta}) = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\mathcal{Z}}$, where $\mathcal{Z} = \int p(\boldsymbol{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. This formulation is reminiscent of the setup proposed by Welling and Teh (2011) and covers many Bayesian machine-learning models, but our formulation has an important difference. SLGD relies on a *diffusion* that reaches the posterior as time approaches infinity. In contrast, our dynamics are *controlled* and the posterior is reached in finite time.

We note that recently submitted concurrent work (Anonymous, 2022) proposes an algorithm akin to ours based on Dai Pra (1991); Tzen and Raginsky (2019b), however their focus and experiments are on estimating the normalising constant of unormalised densities, while our focus is on Bayesian ML tasks such as Bayesian regression, classification and LVMs, thus our work leads to different insights and algorithmic motivations.

1.1. Schrödinger-Föllmer Processes

Let $\mathbb{Q}_0^\gamma$ be the distribution for the solutions to the stochastic differential equation (SDE):

$$d\boldsymbol{\Theta}_t = \boldsymbol{u}_t^0(\boldsymbol{\Theta}_t)dt + \sqrt{\gamma}d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \pi_0^{\mathbb{Q}_0^\gamma}. \tag{2}$$

Definition 1 *(Schrödinger-Bridge Process) The Schrödinger bridge path measure is given by*

$$\mathbb{Q}^* = \inf_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{Q}_0^\gamma\right), \tag{3}$$

where $\mathcal{D}(\pi_0,\pi_1) = \{\mathbb{Q} : (\boldsymbol{\Theta}_0)_\#\mathbb{Q} = \pi_0,\ (\boldsymbol{\Theta}_1)_\#\mathbb{Q} = \pi_1\}$ is the set of path measures with fixed initial and final time-marginals (π_0 at $t = 0$ and π_1 at $t = 1$). Here, $\mathbb{Q}_0^\gamma$ acts as a "prior" and $D_{\mathrm{KL}}(\cdot|\cdot)$ represents the Kullback Leibler (KL) divergence. It is known (Léonard, 2013), that $\mathbb{Q}^*$ is induced by an SDE with modified drift,

$$d\boldsymbol{\Theta}_t = \mathbf{u}_t^*(\boldsymbol{\Theta}_t)dt + \sqrt{\gamma}d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \pi_0, \tag{4}$$

the solution of which is called the Schrödinger-Bridge Process (SBP).

Definition 2 *(Schrödinger-Föllmer Process) The SFP is an SBP where $\pi_0 = \delta_0$ and $\mathbb{Q}_0^\gamma = \mathbb{W}^\gamma$ is the Wiener measure associated to the process described by $d\boldsymbol{\Theta}_t = \sqrt{\gamma}d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0$.*

The SFP differs to the general SBP in that, rather than constraining the initial value to δ_0, the SBP considers *any* initial distribution π_0. The SBP also considers more general Itô SDEs associated with $\mathbb{Q}_0^\gamma$ as the dynamical prior, compared to the SFP which considers only a Wiener process as a prior.

The advantage of considering this more limited version of the SBP is that it admits a closed-form characterisation of the solution to the Schrödinger system (Léonard, 2013; Wang et al., 2021; Pavon et al., 2018), which allows for an unconstrained formulation of the problem. For accessible introductions to the SBP we suggest (Pavon et al., 2018; Vargas et al., 2021). Now we will consider instances of the SBP and the SFP where $\pi_1 = p(\boldsymbol{\theta}|\boldsymbol{X})$.

1.1.1. Analytic Solutions and the Heat Semigroup

Prior work (Pavon, 1989; Dai Pra, 1991; Tzen and Raginsky, 2019b; Huang et al., 2021) has explored the properties of SFPs via a closed form formulation of the Föllmer drift expressed in terms of expectations of Gaussian random variables known as the heat semigroup. The seminal works (Pavon, 1989; Dai Pra, 1991; Tzen and Raginsky, 2019b) highlight how this formulation of the Föllmer drift characterises an exact sampling scheme for a target distribution and how it could potentially be used in practice. The recent work by Huang et al. (2021) builds on Tzen and Raginsky (2019b) and explores estimating the optimal drift in practice via the heat semigroup formulation using a Monte Carlo approximation. Our work aims to take the next step and scale the estimation of the Föllmer drift to high dimensional cases (Graves, 2011; Hoffman et al., 2013). In order to do this we must move away from the heat semigroup and instead consider the dual formulation of the Föllmer drift as a stochastic control problem (Tzen and Raginsky, 2019b).

Definition 3 *The Euclidean heat semigroup Q_t^γ, $t \geq 0$ acts on bounded measurable functions $f : \mathbb{R}^d \to \mathbb{R}$ as $Q_t f(\boldsymbol{x}) = \int_{\mathbb{R}^d} f\left(\boldsymbol{x} + \sqrt{t}\boldsymbol{z}\right) \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \gamma\mathbb{I}) d\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\gamma\mathbb{I})}\left[f\left(\boldsymbol{x} + \sqrt{t}\boldsymbol{z}\right)\right]$.*

In the setting where $\pi_0 = \delta_0$ we can express the optimal SBP drift as follows:

$$\mathbf{u}_t^*(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \ln \mathbb{E}\left[\frac{d\pi_1}{dN(\mathbf{0}, \gamma\mathbb{I})}(\boldsymbol{\Theta}_1)\Big|\boldsymbol{\Theta}_t = \boldsymbol{x}\right], \tag{5}$$

where $\boldsymbol{\Theta}_t$ is the SBP prior $\mathbb{Q}_0^\gamma$. In the SFP case where $\mathbb{Q}_0^\gamma = \mathbb{W}^\gamma$, the optimal drift can be written in terms of the heat semigroup, $\mathbf{u}_t^*(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \ln Q_t^\gamma\left[\frac{d\pi_1}{dN(\mathbf{0},\gamma\mathbb{I})}(\boldsymbol{x})\right]$. Note that an SDE with the above drift, $d\boldsymbol{\Theta}_t^{\mathbf{u}^*} = \nabla_{\boldsymbol{x}} \ln Q_t^\gamma\left[\frac{d\pi_1}{dN(\mathbf{0},\gamma\mathbb{I})}(\boldsymbol{\Theta}_t^{\mathbf{u}^*})\right] dt + \sqrt{\gamma} d\mathbf{B}_t$, satisfies $\text{Law}\boldsymbol{\Theta}_1^{\mathbf{u}^*} = \pi_1$, that is at $t = 1$ these processes are distributed according to our target distribution of interest π_1.

Huang et al. (2021) carried out preliminary work on empirically exploring the success of using the heat semigroup formulation of SFPs in combination with the Euler-Mayurama (EM) discretisation to sample from target distributions in a method they call Schrödinger-Föllmer samplers (SFS). We build on their work by considering a formulation of the Schrödinger-Föllmer process that is suitable for the high dimensional settings arising in Bayesian ML. Our work will focus on a dual formulation of the optimal drift that is closer to variational inference and admits the scalable and flexible parametrisations used in ML.

2. Stochastic Control Formulation

In this section, we introduce a particular formulation of the Schrödinger-Föllmer process in the context of the Bayesian inference problem in Equation 1. In its most general setting of sampling from a target distribution, this formulation was known to Dai Pra (1991). Tzen and Raginsky (2019b) study the theoretical properties of this approach in the context of generative models (Kingma et al., 2021; Goodfellow et al., 2014), finally Opper (2019) applies this formulation to time series modelling. In contrast our focus is on the estimation of a Bayesian posterior for a broader class of models than Tzen and Raginsky explore.

Corollary 4 *The minimiser (where $\mathcal{U}$ is the space of admissible controls)*

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t,\mathbf{\Theta}_t^{\mathbf{u}})||^2 dt - \ln\left(\frac{p(\boldsymbol{X}|\mathbf{\Theta}_1^{\mathbf{u}})p(\mathbf{\Theta}_1^{\mathbf{u}})}{\mathcal{N}(\mathbf{\Theta}_1^{\mathbf{u}}|\mathbf{0},\gamma\mathbb{I}_d)}\right)\right] \tag{6}$$

satisfies $\text{Law}\mathbf{\Theta}_1^{\mathbf{u}^*} = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\mathcal{Z}}$, *where* $d\mathbf{\Theta}_t^{\mathbf{u}} = \mathbf{u}_t(\mathbf{\Theta}_t^{\mathbf{u}})dt + \sqrt{\gamma}d\mathbf{B}_t, \quad \mathbf{\Theta}_0^{\mathbf{u}} \sim \delta_0$.

The objective in Equation 6 can be estimated using an SDE discretisation, such as the EM method. Since the drift $\mathbf{u}_t^*$ is Markov, it can be parametrised by a flexible function estimator such as a neural network, as done in Tzen and Raginsky (2019a,b). In this work we will refer to the above formulation of the SFP as the Neural Schrödinger-Föllmer sampler (NSFS) when we parametrise the drift with a neural network and we implement unbiased mini-batched estimator for this objective detailed in Appendix C.This formulation of SFPs has been previously studied in the context of generative modelling/marginal likelihood estimation (Tzen and Raginsky, 2019b), while we focus on Bayesian inference.

2.1. Theoretical Guarantees for Neural SFS

While the focus in Tzen and Raginsky (2019b) is in providing guarantees in generative models of the form $\boldsymbol{x} \sim q_\phi(\boldsymbol{x}|\boldsymbol{Z}_1)$, $d\boldsymbol{Z}_t = \boldsymbol{u}_\phi(\boldsymbol{Z}_t,t)dt + \sqrt{\gamma}d\mathbf{B}_t,\ \boldsymbol{Z}_0 = 0$, their results extend to our setting as they explore approximating the Föllmer drift for a generic target π_1.

Theorem 4 in Tzen and Raginsky (restated as Theorem 9 in Appendix A.2) motivates using neural networks to parametrise the drift in Equation 6 as it provides a guarantee regarding the expressivity of a network parametrised drift via providing an upper bound on the target distribution error in terms of the size of the network.

We will now proceed to highlight how this error is affected by the EM discretisation:

Corollary 5 *Given the network $\boldsymbol{v}$ from Theorem 9 it follows that the Euler-Mayurama discretisation $\hat{\mathbf{\Theta}}_t^{\boldsymbol{v}}$ of $\mathbf{\Theta}_t^{\boldsymbol{v}}$ has a KL-divergence to the target distribution π_1 of:*

$$D_{\text{KL}}(\pi_1||\hat{\pi}_1^{\boldsymbol{v}}) \leq \left(\epsilon^{1/2} + \mathcal{O}(\sqrt{\Delta t})\right)^2 \tag{7}$$

This result provides us a bound of the error in terms of the depth Δt^{-1} of the stochastic flow and the size of the network that we parametrise the drift with.

2.2. Structured SVI in Models with Local and Global Variables

We consider the general setting where our model has global and local variables $\{\boldsymbol{\theta}_i\}, \mathbf{\Phi}$ satisfying $\boldsymbol{\theta}_i \perp\!\!\!\perp \boldsymbol{\theta}_j|\mathbf{\Phi}$ (Hoffman et al., 2013). This case is particularly challenging as the local variables scale with the size of the dataset and so will the state space. This is a fundamental setting as many hierachical latent variable models in machine learning admit such dependancy structure, such as Topic models (Pritchard et al., 2000; Blei et al., 2003); Bayesian factor analysis (Amari et al., 1996; Bishop, 1999; Klami et al., 2013; Daxberger and Hernández-Lobato, 2019); Variational GP Regression (Hensman et al., 2013); and others.

Remark 6 *The heat semigroup does not preserve conditional independence structure in the drift, i.e. the optimal drift does not decouple and thus depends on the full state-space.*

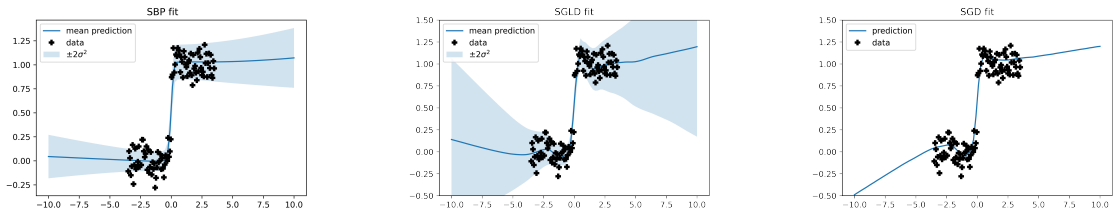

Figure 1: Visual comparison on step function data. We can see how the N-SFS based fits have the best generalisation while SGD and SGLD interpolate the noise.

Remark 6 tells us that the drift is not structured in a way that admits scalable sampling approaches such as stochastic variational inference (SVI) (Hoffman et al., 2013). Additionally this also highlights that the method by Huang et al. (2021) does not scale to models like this as the dimension of the state space will be linear in the size of the dataset.

In a similar fashion to Hoffman and Blei (2015), who focussed on structured SVI, we suggest parametrising the drift via $[\mathbf{u}_t]_{\boldsymbol{\theta}_i} = u^{\boldsymbol{\theta}_i}(t, \boldsymbol{\theta}_i, \Phi, \boldsymbol{x}_i)$; this way the dimension of the drift depends only on their respective local variables and the global variable Φ. While the Föllmer drift does not admit this particular decoupling we can show that this drift is flexible enough to represent general distributions, thus it has the capacity to reach the target distribution. When parametrised in this form we can carry out sampling in the same fashion as SVI whilst maintaining unbiased gradient estimates.

Remark 7 *An SDE parametrised with a decoupled drift* $[\mathbf{u}_t]_{\boldsymbol{\theta}_i} = u^{\boldsymbol{\theta}_i}(t, \boldsymbol{\theta}_i, \Phi, \boldsymbol{x}_i)$ *can reach transition densities which do not factor.*

3. Connections to Variational Inference in Latent Diffusion Models

In this section, we highlight the connection between the objective in Equation 6 to variational inference in models where the latent object is given by an SDE, as studied in Tzen and Raginsky (2019a). Taking inspiration from the recursive nature of Bayesian updates (Khan and Rue, 2021) we develop the following observation.

Lemma 8 *The SBP* $\inf_{\mathbb{Q}\in\mathcal{D}(\delta_0,\, p(\boldsymbol{\theta}|\boldsymbol{X}))} D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{Q}_0^\gamma\right)$ *with reference process* $\mathbb{Q}_0^\gamma$*:*

$$d\boldsymbol{\Theta}_t = \nabla \ln Q_t^\gamma \left[\frac{p(\boldsymbol{\Theta}_t)}{\mathcal{N}(\boldsymbol{\Theta}_t|\mathbf{0}, \gamma\mathbb{I}_d)}\right] + \sqrt{\gamma} d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0, \tag{8}$$

corresponds to maximising the ELBO of the model:

$$\mathbf{x}_i \sim p(\mathbf{x}_i|\boldsymbol{\Theta}_1), \quad d\boldsymbol{\Theta}_t = \nabla \ln Q_t^\gamma \left[\frac{p(\boldsymbol{\Theta}_t)}{\mathcal{N}(\boldsymbol{\Theta}_t|\mathbf{0}, \gamma\mathbb{I}_d)}\right] + \sqrt{\gamma} d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0.$$

In short we can view a variant of the objective in Equation 6 as an instance of variational Bayesian inference with an SDE prior. Note that this provides a succinct connection between variational inference and maximum entropy in path space (Léonard, 2012).

Table 1: a9a dataset.

Method	Accuracy	ECE	Log Likelihood
N-SFS	0.8498 ± 0.0002	0.0099 ± 0.001	-0.3407 ± 0.0004
SGLD	0.8515 ± 0.001	0.001 ± 0.002	-0.3247 ± 0.0002

Table 2: Aug-MNIST (Ferianc et al., 2021).

Method	Accuracy	ECE	Log Likelihood
N-SFS	0.9479 ± 0.0043	0.0077 ± 0.0012	-0.3890 ± 0.0374
SGLD	0.9247 ± 0.0035	0.0141 ± 0.0018	-0.2439 ± 0.0118
SGD	0.9404 ± 0.0031	0.0284 ± 0.0021	-

Table 3: Step function dataset.

Method	MSE	Log Likelihood
N-SFS	0.0028 ± 0.00097	-63.048 ± 8.2760
SGLD	0.1774 ± 0.128	-1389.581 ± 834.968

Table 4: MEG dataset.

Method	Log Likelihood
N-SFS	-5.110972 ± 0.128856
SGLD	-4.936021 ± 0.042283

4. Experimental Results

We ran experiments on Bayesian NN regression, classification, logistic regression and ICA (Amari et al., 1996), reporting accuracies, log joints (Welling and Teh, 2011; Izmailov et al., 2021) and expected calibration error (ECE) (Guo et al., 2017). For details on exact experimental setups please see Appendix F. Across experiments we compare to SGLD as it has been shown to be a competitive baseline in Bayesian deep learning (Izmailov et al., 2021). In the Bayesian NN tasks the likelihood is parametrised via $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = p(\boldsymbol{y}_i|f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))$ where $f_{\boldsymbol{\theta}}$ is a neural network and the prior on $\boldsymbol{\theta}$ is Gaussian.

Step Function: We fit a 2-hidden-layer neural network with a total of 14876 parameters on a toy step function dataset (see Figure 1). We can see in Figure how both the SGD and SGLD fits interpolate the noise, whilst N-SFS has straight lines and thus achieves a better test error, whilst having well calibrated error bars. We believe it is a great milestone to see how an overparametrised neural network is able to achieve such well calibrated predictions.

Augmented MNIST:We train the standard LeNet5 (LeCun et al., 1998) architecture (with 44426 parameters) on the MNIST dataset (LeCun and Cortes, 2010). At test time we evaluate the methods on the MNIST test set augmented by random rotations of up to 30°(Ferianc et al., 2021). Table 2 shows how N-SFS has the highest accuracy whilst obtaining the lowest calibration error among the considered methods, highlighting that our approach has the most well calibrated and accurate predictions when considering a slightly perturbed test set. We highlight how LeNet5 falls into an interesting regime as the number of parameters is considerably less than the size of the training set, and thus we can argue it is not in the overparametrised regime. This regime (Belkin et al., 2019) has been shown to be challenging in achieving good generalisation errors, thus we believe the predictive and calibrated accuracy achieved by N-SFS is a strong milestone.

a9a/MEG Datasets: Following Welling and Teh (2011) we explore a logistic regression model on the a9a dataset and a Bayesian variant of ICA on the MEG-Dataset (Vigario, 1997). We can observe (Tables 4, 1) that N-SFFS achieves results comparable to SGLD.

5. Discussion and Future Directions

We are competitive to SGLD and obtain better calibrated predictions in high-dimensional (Bayesian DL) settings. We would like to highlight that these results were achieved without

any tuning and simple NN architectures. We believe that our results illustrate how stochastic control objectives constitute a promising and exciting avenue for approximate Bayesian inference. We are currently exploring the decoupled drift parametrisations (Section 2.2) for models such as LDA as well as bench-marking the performance of different estimators such as STL (Xu et al., 2021) and VarGrad (Nüsken and Richter, 2021). Additionally we notice that the architecture used in the drift network can influence results thus we will explore modern architectures for diffusions such as the score networks in De Bortoli et al. (2021).

Acknowledgements. Francisco Vargas is Funded by Huawei Technologies Co. This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through the grant CRC 1114 'Scaling Cascades in Complex Systems' (project A02, project number 235221301). Andrius Ovsianas is funded by EPSRC iCASE Award EP/T517677/1. Mark Girolami is supported by a Royal Academy of Engineering Research Chair, and EPSRC grants EP/T000414/1, EP/R018413/2, EP/P020720/2, EP/R034710/1, EP/R004889/1.

References

Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. A new learning algorithm for blind signal separation. In *Advances in neural information processing systems*, pages 757–763. Morgan Kaufmann Publishers, 1996.

Anonymous. Path integral sampler: A stochastic control approach for sampling. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y. under review.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E Jacob. Schrödinger bridge samplers. *arXiv preprint*, 2019.

Christopher M Bishop. Bayesian PCA. *Advances in neural information processing systems*, pages 382–388, 1999.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Michelle Boué and Paul Dupuis. A variational representation for certain functionals of Brownian motion. *The Annals of Probability*, 26(4):1641–1659, 1998.

Lenaic Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. Faster Wasserstein distance estimation with the Sinkhorn divergence. *Advances in Neural Information Processing Systems*, 33, 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991.

Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *arXiv preprint arXiv:2106.01357*, 2021.

Martin Ferianc, Partha Maji, Matthew Mattina, and Miguel Rodrigues. On the effects of quantisation on model uncertainty in Bayesian neural networks. *arXiv preprint arXiv:2102.11062*, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

István Gyöngy and Nicolai Krylov. Existence of strong solutions for Itô's stochastic equations via approximations. *Probability theory and related fields*, 105(2):143–158, 1996.

Carsten Hartmann and Christof Schütte. Efficient rare event simulation by optimal nonequilibrium forcing. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(11): P11004, 2012.

Carsten Hartmann, Lorenz Richter, Christof Schütte, and Wei Zhang. Variational characterization of free energy: Theory and algorithms. *Entropy*, 19(11):626, 2017.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Matthew D Hoffman and David M Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pages 361–369, 2015.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

Jian Huang, Yuling Jiao, Lican Kang, Xu Liao, Jin Liu, and Yanyan Liu. Schrödinger-Föllmer sampler: Sampling without ergodicity. *arXiv preprint arXiv:2106.10880*, 2021.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are Bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021.

Hilbert J Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201, 2005.

Mohammad Emtiyaz Khan and Håvard Rue. The bayesian learning rule. *arXiv preprint arXiv:2107.04562*, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14(4), 2013.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Christian Léonard. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4):1879–1920, 2012.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Dimitra Maoutsa and Manfred Opper. Deterministic particle flows for constraining SDEs. *arXiv preprint arXiv:2110.13020*, 2021.

Nikolas Nüsken and Lorenz 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*, 2(4):1–48, 2021.

Manfred Opper. Variational inference for stochastic differential equations. *Annalen der Physik*, 531(3):1800233, 2019.

Michele Pavon. Stochastic control and nonequilibrium thermodynamical systems. *Applied Mathematics and Optimization*, 19(1):187–202, 1989.

Michele Pavon, Esteban G. Tabak, and Giulio Trigila. The data-driven Schrödinger bridge. *arXiv preprint*, 2018.

JK Pritchard, Stephens M., and Donnelly P. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Sebastian Reich. Data assimilation: the Schrödinger perspective. *Acta Numerica*, 28:635–711, 2019.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco JR Ruiz, and Ömer Deniz Akyildiz. Vargrad: a low-variance gradient estimator for variational inference. *arXiv preprint arXiv:2010.10436*, 2020.

Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. *arXiv preprint arXiv:1703.09194*, 2017.

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932.

Sep Thijssen and HJ Kappen. Path integral control and state-dependent feedback. *Physical Review E*, 91(3):032104, 2015.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019a.

Belinda Tzen and Maxim Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pages 3084–3114. PMLR, 2019b.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving Schrödinger bridges via maximum likelihood. *Entropy*, 23(9), 2021. ISSN 1099-4300. doi: 10.3390/e23091134. URL https://www.mdpi.com/1099-4300/23/9/1134.

Ricardo Vigario. Meg data for studies using independent component analysis. *http://www. cis. hut. fi/projects/ica/eegmeg/MEG_data. html*, 1997.

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via Schrödinger bridge. *arXiv preprint arXiv:2106.10410*, 2021.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Winnie Xu, Ricky T. Q. Chen, Xuechen Li, and David Duvenaud. Infinitely deep Bayesian neural networks with stochastic differential equations. *arXiv preprint arXiv:2102.06559*, 2021.

Appendix A. Main Results

A.1. Posterior Drift

Corollary 4 *The minimiser*

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t,\mathbf{\Theta}_t)||^2 dt - \ln\left(\frac{p(\boldsymbol{X}|\mathbf{\Theta}_1)p(\mathbf{\Theta}_1)}{\mathcal{N}(\mathbf{\Theta}_1|\mathbf{0},\gamma\mathbb{I}_d)}\right)\right] \tag{9}$$

satisfies $\mathrm{Law}\mathbf{\Theta}_1^{\mathbf{u}^*} = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\mathcal{Z}}$, *where*

$$d\mathbf{\Theta}_t = \sqrt{\gamma}d\mathbf{B}_t, \quad \mathbf{\Theta}_0 \sim \delta_0, \tag{10}$$

and

$$d\mathbf{\Theta}_t^{\mathbf{u}^*} = \mathbf{u}^*(t,\mathbf{\Theta}_t)dt + \sqrt{\gamma}d\mathbf{B}_t, \quad \mathbf{\Theta}_0^{\mathbf{u}^*} \sim \delta_0. \tag{11}$$

Proof This follows directly after substituting the Radon-Nikodym derivative between the Gaussian distribution and the posterior into Theorem 1 in Tzen and Raginsky (2019b) or Theorem 3.1 in Dai Pra (1991). ■

A.2. EM-Discretisation Result

First we would like to introduce the following auxiliary theorem from Tzen and Raginsky (2019b):

Theorem 9 *(Tzen and Raginsky, 2019b) Given the standard regularity assumptions presented for* $f = \frac{d\pi_1}{d\mathcal{N}(\mathbf{0},\gamma\mathbb{I})}$ *in Tzen and Raginsky (2019b), let* $L = \max\{\mathrm{Lip}(f), \mathrm{Lip}(\nabla f)\}$ *and assume that there exists a constant* $c \in (0,1]$ *such that* $f \geq c$*. Then for any* $0 < \epsilon < 16\frac{L^2}{c^2}$ *there exists a neural net* $\boldsymbol{v} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ *with size polynomial in* $1/\epsilon, d, L, c, 1/c, \gamma$*, such that the activation function of each neuron follows the regularity assumptions in Tzen and Raginsky (2019b) (e.g.* $\mathrm{ReLU}, \mathrm{Sigmoid}, \mathrm{Softplus}$*) and*

$$D_{\mathrm{KL}}(\pi_1 || \pi_1^{\boldsymbol{v}}) \leq \epsilon, \tag{12}$$

where $\pi_1^{\boldsymbol{v}} = \mathrm{Law}(\boldsymbol{\Theta}_1^{\boldsymbol{v}})$ *is the terminal distribution of the the diffusion process*

$$d\boldsymbol{\Theta}_t^{\boldsymbol{v}} = \boldsymbol{v}(\boldsymbol{\Theta}_t^{\boldsymbol{v}}, \sqrt{1-t})dt + \sqrt{\gamma}d\mathbf{B}_t, \qquad t \in [0,1]. \tag{13}$$

We can now proceed to prove the direct corollary of the above theorem when using the EM scheme for simulation.

Corollary 5 *Given the network* $\boldsymbol{v}$ *from Theorem 9 it follows that the Euler-Mayurama discretisation* $\hat{X}_t^{\boldsymbol{v}}$ *of* $X_t^{\boldsymbol{v}}$ *has a KL-divergence to the target distribution* π_1 *of:*

$$D_{\mathrm{KL}}(\pi_1 || \hat{\pi}_1^{\boldsymbol{v}}) \leq \left(\epsilon^{1/2} + \mathcal{O}(\sqrt{\Delta t})\right)^2 \tag{14}$$

Proof Consider the path-wise KL between the exact Schrödinger-Föllmer process and its EM-discretised neural approximation:

$$D_{\mathrm{KL}}(\mathbb{P}^{\boldsymbol{u}^*} || \mathbb{P}^{\hat{\boldsymbol{v}}}) = \frac{1}{2\gamma}\int_0^1 \mathbb{E}||\boldsymbol{u}^*(\boldsymbol{\Theta}_t^{\boldsymbol{u}^*}, t) - \hat{\boldsymbol{v}}(\boldsymbol{\Theta}_t^{\boldsymbol{u}^*}, \sqrt{1-t})||^2 dt. \tag{15}$$

Defining $d(\boldsymbol{x}, \boldsymbol{y}) := \sqrt{\frac{1}{2\gamma}\int_0^1 \mathbb{E}||\boldsymbol{x}(\boldsymbol{\Theta}_t^{\boldsymbol{u}^*}, t) - \hat{\boldsymbol{y}}(\boldsymbol{\Theta}_t^{\boldsymbol{u}^*}, t)||^2 dt}$, it is clear that $d(\boldsymbol{x}, \boldsymbol{y})$ satisfies the triangle inequality as it is the $\mathcal{L}^2(\mathbb{Q}^{\mathbf{u}^*})$ metric between drifts, thus applying the triangle inequality at the drift level we have that (for simplicitly let $\gamma = 1$):

$$d(\boldsymbol{u}^*, \hat{\boldsymbol{v}}) \leq \left(\int_0^1 \mathbb{E}\left[||\boldsymbol{u}_t^* - \boldsymbol{v}_{\sqrt{1-t}}||^2\right] dt\right)^{1/2} + \left(\int_0^1 \mathbb{E}\left|||\boldsymbol{v}_{\sqrt{1-t}} - \hat{\boldsymbol{v}}_{\sqrt{1-t}}||^2\right] dt\right)^{1/2}$$

From Tzen and Raginsky (2019b) we can bound the first term resulting in:

$$d(\boldsymbol{u}^*, \hat{\boldsymbol{v}}) \leq \epsilon^{1/2} + \left(\int_0^1 \mathbb{E}\left[||\boldsymbol{v}_{\sqrt{1-t}} - \hat{\boldsymbol{v}}_{\sqrt{1-t}}||^2\right] dt\right)^{1/2}$$

Now remembering that the EM drift is given by $\hat{\boldsymbol{v}}_{\sqrt{1-t}}(\boldsymbol{\Theta}_t) = \boldsymbol{v}(\hat{\boldsymbol{\Theta}}_t, \sqrt{1-\Delta t\lceil t/\Delta t\rceil})$, we can use that $\boldsymbol{v}$ is L'-Lipschitz in both arguments thus:

$$\begin{aligned} d(\boldsymbol{u}^*, \hat{\boldsymbol{v}}) &\leq \epsilon^{1/2} + \left(L'^2 \int_0^1 \left(\mathbb{E}\left[(||\boldsymbol{\Theta}_t^{\boldsymbol{u}^*} - \hat{\boldsymbol{\Theta}}_t^{\boldsymbol{u}^*}|| + \Delta t)^2\right]\right) dt\right)^{\frac{1}{2}} \\ &\leq \epsilon^{1/2} + \left(2L'^2 \left(\mathbb{E}\left[\int_0^1 ||\boldsymbol{\Theta}_t^{\boldsymbol{u}^*} - \hat{\boldsymbol{\Theta}}_t^{\boldsymbol{u}^*}||^2 dt\right] + \Delta t^2\right)\right)^{\frac{1}{2}} \\ &\leq \epsilon^{1/2} + \left(2L'^2 \left(\mathbb{E}\left[\max_{0\leq t\leq 1} ||\boldsymbol{\Theta}_t^{\boldsymbol{u}^*} - \hat{\boldsymbol{\Theta}}_t^{\boldsymbol{u}^*}||^2\right] + \Delta t^2\right)\right)^{\frac{1}{2}} \end{aligned}$$

which using the strong convergence of the EM approximation (Gyöngy and Krylov, 1996) implies:

$$\mathbb{E}\left[\max_{0\leq t\leq 1} ||\boldsymbol{\Theta}_t^{\boldsymbol{v}^*} - \hat{\boldsymbol{\Theta}}_t^{\boldsymbol{v}^*}||^2\right] \leq C_{L'}\Delta t, \tag{16}$$

thus:

$$d(\boldsymbol{u}^*, \hat{\boldsymbol{v}}) \leq \epsilon^{1/2} + L'\sqrt{2}\left(\sqrt{C_{L'}\Delta t} + \Delta t\right),$$

squaring both sides and applying the data processing inequality completes the proof. ■

Appendix B. Connections to VI

We first start by making the connection in a simpler case – when the prior of our Bayesian model is given by a Gaussian distribution with variance γ, that is $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \gamma\mathbb{I}_d)$.

Observation 1 *When $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \gamma\mathbb{I}_d)$, it follows that the N-SFP objective in Equation 6 corresponds to the negative ELBO of the model:*

$$\begin{aligned} d\boldsymbol{\Theta}_t &= \sqrt{\gamma} d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0, \\ \mathbf{x}_i &\sim p(\mathbf{x}_i|\boldsymbol{\Theta}_1). \end{aligned} \tag{17}$$

Proof Substituting $p(\boldsymbol{\theta})$ into Equation 6 yields

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}_t||^2 dt - \ln p(\boldsymbol{X}|\boldsymbol{\Theta}_1)\right]. \tag{18}$$

Then, from (Boué and Dupuis, 1998; Tzen and Raginsky, 2019a,b) we know that the term $\mathbb{E}\left[\int_0^1 ||\mathbf{u}_t||^2 dt - \ln p(\boldsymbol{X}|\boldsymbol{\Theta}_1)\right]$ is the negative ELBO of the model specified in Equation 17. ■

While the above observation highlights a specific connection between N-SFP and VI, it is limited to Bayesian models that are specified with Gaussian priors. To extend the result, we take inspiration from the recursive nature of Bayesian updates in the following result.

Lemma 8 *The SBP*

$$\inf_{\mathbb{Q}\in\mathcal{D}(\delta_0,\, p(\boldsymbol{\theta}|\boldsymbol{X}))} D_{\mathrm{KL}}\left(\mathbb{Q}\big|\big|\mathbb{Q}_0^{\gamma}\right), \tag{19}$$

with reference process $\mathbb{Q}_0^{\gamma}$*:*

$$d\boldsymbol{\Theta}_t = \nabla \ln Q_t^{\gamma}\left[\frac{p(\boldsymbol{\Theta}_t)}{\mathcal{N}(\boldsymbol{\Theta}_t|\mathbf{0},\gamma\mathbb{I}_d)}\right] + \sqrt{\gamma}d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0, \tag{20}$$

corresponds to maximising the ELBO of the model:

$$\begin{aligned} d\boldsymbol{\Theta}_t &= \nabla \ln Q_t^{\gamma}\left[\frac{p(\boldsymbol{\Theta}_t)}{\mathcal{N}(\boldsymbol{\Theta}_t|\mathbf{0},\gamma\mathbb{I}_d)}\right] + \sqrt{\gamma}d\mathbf{B}_t, \quad \boldsymbol{\Theta}_0 \sim \delta_0, \\ \mathbf{x}_i &\sim p(\mathbf{x}_i|\boldsymbol{\Theta}_1). \end{aligned} \tag{21}$$

Proof For brevity let $\boldsymbol{u}_t^0(\boldsymbol{\theta}) = \nabla \ln Q_t^{\gamma}\left[\frac{p(\boldsymbol{\theta})}{\mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\gamma\mathbb{I}_d)}\right]$. First notice that the time-one marginals of $\mathbb{Q}_0^{\gamma}$ are given by the Bayesian prior:

$$(\boldsymbol{\Theta}_1)_{\#}\mathbb{Q}_0^{\gamma} = p(\boldsymbol{\theta})$$

Now from Léonard (2012); Pavon et al. (2018) we know that the Schrödinger system is given by:

$$\phi_0(\boldsymbol{\theta}_0)\int p(\boldsymbol{\theta}_0, 0, y, \boldsymbol{\theta}_1)\hat{\phi}_1(\boldsymbol{\theta}_1)d\boldsymbol{\theta}_1 = \delta_0(\theta), \tag{22}$$

$$\hat{\phi}_1(\boldsymbol{\theta}_1)\int p(\boldsymbol{\theta}_0, 0, \boldsymbol{\theta}_1, 1)\phi_0(\boldsymbol{\theta}_0)d\boldsymbol{\theta}_0 = p(\boldsymbol{\theta}_1|\boldsymbol{X}), \tag{23}$$

where Equation 22 can be given a rigorous meaning in weak form (that is, by integrating against suitable test functions). Notice $\phi_0 = \delta_0$ and thus it follows that:

$$\hat{\phi}_1(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}|\boldsymbol{X})}{p(0,0,\boldsymbol{\theta},1)} = \frac{p(\boldsymbol{\theta}|\boldsymbol{X})}{p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})}{\mathcal{Z}}, \tag{24}$$

then by Pavon (1989); Dai Pra (1991); Pavon et al. (2018) the optimal drift is given by:

$$\mathbf{u}_t^*(\boldsymbol{\theta}) = \gamma\nabla \ln \mathbb{E}[p(\boldsymbol{X}|\boldsymbol{\Theta}_1)|\boldsymbol{\Theta}_t = \boldsymbol{\theta}], \tag{25}$$

where the expectation is taken with respect to the reference process $\mathbb{Q}_0^{\gamma}$. Now if we let $v(\theta, t) = -\ln \mathbb{E}[p(\boldsymbol{X}|\boldsymbol{\Theta}_1)|\boldsymbol{\Theta}_t = \boldsymbol{\theta}]$ be our value function then via the linearisation of the Hamilton-Bellman-Jacobi Equation through Fleming's logarithmic transform (Kappen, 2005; Thijssen and Kappen, 2015; Tzen and Raginsky, 2019b) it follows that said value function satisfies:

$$v(\boldsymbol{\theta}, t) = \min_{\mathbf{u}\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2\gamma}\int_t^1 ||\mathbf{u}_t - \mathbf{u}_t^0||^2 dt - \ln p(\boldsymbol{X}|\boldsymbol{\Theta}_1)\Big|\boldsymbol{\Theta}_t = \boldsymbol{\theta}\right] \tag{26}$$

and thus $\mathbf{u}_t^*(\boldsymbol{\theta}) = \gamma\nabla \ln \mathbb{E}[p(\boldsymbol{X}|\boldsymbol{\Theta}_1)|\boldsymbol{\Theta}_t = \boldsymbol{\theta}]$ is a minimiser to:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{U}} \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}_t - \mathbf{u}_t^0||^2 dt - \ln p(\boldsymbol{X}|\boldsymbol{\Theta}_1)\right]. \tag{27}$$

■

Note Lemma 8 induces a new method in which we first estimate a prior reference process's as in Equation 8 and then we optimise the ELBO for the model in 9, this raises the question on what effect the dynamical prior can have within SBP based frameworks.

Algorithm 1: Optimization of N-SFS

Data: data set $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^N$, initialized drift neural network u_ϕ, parameter dimension d, number of iterations M, batch size B, number of Euler-Maruyama discretization steps k, diffusion coefficient γ.

$\Delta t \leftarrow \frac{1}{k}$

$t_j \leftarrow j\Delta t$ for all $j = 0, \dots, k$

for $i = 1, \dots, M$ **do**

  Initialize $\boldsymbol{\Theta}_0^s \leftarrow 0 \in \mathbb{R}^d$ for all $s = 1, \dots, S$

  $\{\Theta_j^{s\phi}\}_{j=1}^k \leftarrow$ Euler-Maruyama$(u_\phi, \boldsymbol{\Theta}_0^s, \Delta t)$ for all $s = 1, \dots, S$

  Sample $\mathbf{x}_{r_1}, \dots, \mathbf{x}_{r_B} \sim \boldsymbol{X}$

  Compute

  $g \leftarrow \nabla_\phi \left(\frac{1}{S} \sum_{s=1}^S \sum_{j=0}^k \left(||u_\phi(\boldsymbol{\Theta}_j^{s\phi}, t_j)||^2 - \ln\left(\frac{p(\boldsymbol{\Theta}_k^{s\phi})}{\mathcal{N}(\boldsymbol{\Theta}_k^{s\phi}|\mathbf{0}, \gamma\mathbb{I}_d)} \right) + \frac{N}{B} \sum_{j=1}^B \ln p(\mathbf{x}_{r_j}|\boldsymbol{\Theta}_k^{s\phi}) \right) \right)$

  $\phi \leftarrow$ Gradient Step(ϕ, g)

end

return u_ϕ

Appendix C. Stochastic Variational Inference

For a Bayesian model having the structure specified by equation 1 the objective in equation 6 can be written as follows:

$$\mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t, \boldsymbol{\Theta}_t^{\mathbf{u}})||^2 dt - \ln\left(\frac{p(\boldsymbol{X}|\boldsymbol{\Theta}_1^{\mathbf{u}})p(\boldsymbol{\Theta}_1^{\mathbf{u}})}{\mathcal{N}(\boldsymbol{\Theta}_1^{\mathbf{u}}|\mathbf{0}, \gamma\mathbb{I}_d)}\right)\right] = \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t, \boldsymbol{\Theta}_t^{\mathbf{u}})||^2 dt - \ln\left(\frac{p(\boldsymbol{\Theta}_1^{\mathbf{u}})}{\mathcal{N}(\boldsymbol{\Theta}_1^{\mathbf{u}}|\mathbf{0}, \gamma\mathbb{I}_d)}\right)\right] \tag{28}$$

$$+ \sum_{i=1}^N \mathbb{E}[\ln p(\mathbf{x}_i|\boldsymbol{\Theta}_1^{\mathbf{u}})] \tag{29}$$

where the last term can be written as:

$$\sum_{i=1}^N \mathbb{E}[\ln p(\mathbf{x}_i|\boldsymbol{\Theta}_1^{\mathbf{u}})] = \frac{N}{B}\mathbb{E}_{\mathbf{x}_{k_i}\sim\mathcal{D}}\left[\sum_{i=1}^B \mathbb{E}\left[\ln p(\mathbf{x}_{k_i}|\boldsymbol{\Theta}_1^{\mathbf{u}})\right]\right] \tag{30}$$

That is, it is possible to estimate the objective (and its gradients) by subsampling the data with random batches of size B and using the scaling $\frac{N}{B}$. A version of the algorithm with Euler-Maruyama discretization of the SDE is given in Algorithm 1.

Appendix D. Decoupled Drift Results

First let us consider the setting where the local variables are fully independent, that is, $\boldsymbol{\theta}_i \perp\!\!\!\perp \boldsymbol{\theta}_j$.

Remark 10 *The heat semigroup preserves fully factored (mean-field) distributions thus the Föllmer drift is decoupled.*

In this setting we can parametrise the dimensions of the drift which correspond to local variables in a decoupled manner, $[\mathbf{u}_t]_{\boldsymbol{\theta}_i} = u^{\boldsymbol{\theta}_i}(t, \boldsymbol{\theta}_i, \boldsymbol{x}_i)$. This amortised parametrisation (Kingma and Welling, 2013) allows us to carry out gradient estimates using a mini-batch (Hoffman et al., 2013) rather than hold the whole state space in memory.

Remark 6 *The heat semigroup does not preserve conditional independence structure in the drift. That is, the optimal drift does not decouple and as a result depends on the full state space.*

Proof Consider the following distribution:

$$\mathcal{N}(x|z,0)\mathcal{N}(y|z,0)\mathcal{N}(z|0,1) \tag{31}$$

We want to estimate:

$$\mathbb{E}\left[\frac{\mathcal{N}(X+x|Z+z,1)\mathcal{N}(Y+y|Z+z,1)\mathcal{N}(Z+z|1,0)}{\mathcal{N}(X+x|0,1)\mathcal{N}(Y+y|0,1)\mathcal{N}(Z+z|0,1)}\right] \tag{32}$$

where $X, Y, Z \sim \mathcal{N}(0, \sqrt{1-t})$

$$\mathbb{E}\left[\frac{\mathcal{N}(X+x|Z+z,1)\mathcal{N}(Y+y|Z+z,1)}{\mathcal{N}(X+x|0,1)\mathcal{N}(Y+y|0,1)}\right] \tag{33}$$

we can easily see that the above no longer has conditional independence structure and thus when taking its logarithmic derivative the drift does not decouple. ■

Remark 7 *An SDE parametrised with a decoupled drift $[\mathbf{u}_t]_{\boldsymbol{\theta}_i} = u(t, \boldsymbol{\theta}_i, \Phi, \boldsymbol{x}_i)$ can reach transition densities which do not factor.*

Proof Consider the linear time-homogeneous SDE:

$$d\boldsymbol{\Theta}_t = \boldsymbol{A}\boldsymbol{\Theta}_t dt + \gamma d\boldsymbol{W}_t, \quad \boldsymbol{\Theta}_0 = 0, \tag{34}$$

where:

$$[\boldsymbol{A}]_{ij} = \delta_{ij} + i\delta_{1j}, \tag{35}$$

then this SDE admits a closed form solution:

$$\boldsymbol{\Theta}_t = \gamma \int_0^t \exp\left(\boldsymbol{A}(t-s)\right) d\boldsymbol{W}_s, \tag{36}$$

which is a Gauss-Markov process with 0 mean and covariance matrix:

$$\boldsymbol{\Sigma}(t) = \gamma^2 \int_0^t \exp\left(\boldsymbol{A}(t-s)\right) \exp\left(\boldsymbol{A}(t-s)\right)^\top ds \tag{37}$$

We can carry out the matrix exponential through the eigendecomposition of $\boldsymbol{A}$, for simplicity let us consider the 3-dimensional case:

$$\exp\left(\boldsymbol{A}(t-s)\right) = Se^{D(t-s)}S^{-1} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} e^{t-s} & 0 & 0 \\ 0 & e^{t-s} & 0 \\ 0 & 0 & e^{3(t-s)} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1/2 \\ 0 & 0 & 1/2 \end{pmatrix} \tag{38}$$

From this we see that:

$$\begin{aligned} \exp\left(\boldsymbol{A}(t-s)\right)\exp\left(\boldsymbol{A}(t-s)\right)^\top &= Se^{D(t-s)}S^{-1}(Se^{D(t-s)}S^{-1})^\top && (39)\\ &= Se^{D(t-s)}S^{-1}S^{-\top}e^{D(t-s)}S^\top && (40)\\ &= \frac{1}{4}Se^{D(t-s)}\begin{pmatrix} 8 & 2 & -2 \\ 2 & 5 & -1 \\ -2 & -1 & 1 \end{pmatrix} e^{D(t-s)}S^\top && (41)\\ &= \frac{1}{4}S\begin{pmatrix} 8e^{2(t-s)} & 2e^{2(t-s)} & -2e^{4(t-s)} \\ 2e^{2(t-s)} & 5e^{2(t-s)} & -e^{4(t-s)} \\ -2e^{4(t-s)} & -e^{4(t-s)} & e^{6(t-s)} \end{pmatrix} S^\top && (42) \end{aligned}$$

Integrating wrt to s yields:

$$\begin{aligned} \int \exp\left(\boldsymbol{A}(t-s)\right)\exp\left(\boldsymbol{A}(t-s)\right)^\top ds &= \frac{1}{4}S\begin{pmatrix} 4 & 1 & -\frac{1}{2} \\ 1 & \frac{5}{2} & -\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{4} & \frac{1}{6} \end{pmatrix} S^\top && (43)\\ &= \frac{1}{24}\begin{pmatrix} 13 & 2 & -1 \\ 2 & 16 & -2 \\ -1 & -2 & 4 \end{pmatrix}. && (44) \end{aligned}$$

The covariance matrix is dense at all times and thus the density $\text{Law}(\boldsymbol{\Theta}_t) = \mathcal{N}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t))$ does not factor (is a fully joint distribution). This example motivates that even with the decoupled drift we can reach coupled distributions. ■

Appendix E. Sticking the Landing and Low Variance Estimators

As with VI (Richter et al., 2020; Roeder et al., 2017), the gradient of the objective in this study admits several low variance estimators (Nüsken and Richter, 2021; Xu et al., 2021). In this section we formally recap what it means for an estimator to "stick the landing" and we prove that the estimator proposed in Xu et al. satisfies said property.

The full objective being minimised in our approach is:

$$J(\mathbf{u}) = \mathbb{E}[\mathcal{F}(\mathbf{u})] = \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t,\boldsymbol{\Theta}_t)||^2 dt + \frac{1}{\sqrt{\gamma}}\int_0^1 \mathbf{u}(t,\boldsymbol{\Theta}_t)^\top d\mathbf{B}_t - \ln\left(\frac{p(\boldsymbol{X}|\boldsymbol{\Theta}_1)p(\boldsymbol{\Theta}_1)}{\mathcal{N}(\boldsymbol{\Theta}_1|\mathbf{0},\gamma\mathbb{I}_d)}\right)\right], \tag{45}$$

noticing that in previous formulations we have omitted the Itô integral as it has zero expectation (but the integral appears naturally through Girsanov's theorem). We call the

estimator calculated by taking gradients of the above objective the relative-entropy estimator. The estimator proposed in Xu et al. (2021) (Sticking the landing estimator) is given by:

$$J_{\mathrm{STL}}(\mathbf{u}) = \mathbb{E}[\mathcal{F}_{\mathrm{STL}}(\mathbf{u})] = \mathbb{E}\left[\frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t,\mathbf{\Theta}_t)||^2 dt + \frac{1}{\sqrt{\gamma}}\int_0^1 \mathbf{u}^{\perp}(t,\mathbf{\Theta}_t)^{\top} d\mathbf{B}_t - \ln\left(\frac{p(\boldsymbol{X}|\mathbf{\Theta}_1)p(\mathbf{\Theta}_1)}{\mathcal{N}(\mathbf{\Theta}_1|\mathbf{0},\gamma\mathbb{I}_d)}\right)\right], \tag{46}$$

where $\perp$ means that the gradient is stopped/detached as in Xu et al. (2021); Roeder et al. (2017).

We study perturbations of $\mathcal{F}$ around $\mathbf{u}^*$ by considering $\mathbf{u}^* + \varepsilon\boldsymbol{\phi}$, with $\boldsymbol{\phi}$ arbitrary, and ε small. More precisely, we set out to compute:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0}, \tag{47}$$

through which we define the definition of sticking the landing:

Definition 11 *We say that an estimator sticks the landing when*

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0} = 0, \tag{48}$$

almost surely, for all smooth and bounded perturbations $\boldsymbol{\phi}$.

Notice that by construction, $\boldsymbol{u}^*$ is a global minimiser of J, and hence all directional derivatives vanish,

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathbb{E}[\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})]\Big|_{\varepsilon=0} = 0. \tag{49}$$

Definition 11 additionally demands that this quantity is zero almost surely, and not just on average. Consequently, "sticking the landing"-estimators will have zero-variance at $\boldsymbol{u}^*$.

Remark 12 *The relative-entropy stochastic control estimator does not stick the landing.*

Proof See Nüsken and Richter (2021), Theorem 5.3.1, clause 3, Equation 133 clearly indicates $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0} \neq 0$. ■

We can now go ahead and prove that the estimator proposed by Xu et al. (2021) does indeed stick the landing.

Theorem 13 *The STL estimator proposed in (Xu et al., 2021) satisfies:*

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0} = 0, \tag{50}$$

almost surely, for all smooth and bounded perturbations $\boldsymbol{\phi}$.

Proof Let us decompose $\mathcal{F}$ in the following way:

$$\mathcal{F}(\mathbf{u}) = \mathcal{F}_0(\mathbf{u}) + \mathcal{F}_1(\mathbf{u}) \tag{51}$$

where (denoting the terminal cost with g):

$$\mathcal{F}_0(\mathbf{u}) = \frac{1}{2\gamma}\int_0^1 ||\mathbf{u}(t, \mathbf{\Theta}_t)||^2 dt + g(\mathbf{\Theta}_1) \tag{52}$$

$$\mathcal{F}_1(\mathbf{u}) = \frac{1}{\sqrt{\gamma}}\int_0^1 \mathbf{u}^{\perp}(t, \mathbf{\Theta}_t)^{\top} d\mathbf{B}_t \tag{53}$$

From Nüsken and Richter (2021), Theorem 5.3.1, Equation 133 it follows that:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}_0(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\bigg|_{\varepsilon=0} = -\frac{1}{\sqrt{\gamma}}\int_0^1 \boldsymbol{A}_t \cdot (\nabla \mathbf{u}_t^*)(\boldsymbol{\theta}_t^{\mathbf{u}^*})\,\mathrm{d}\mathbf{B}_t, \tag{54}$$

almost surely, where $\boldsymbol{A}_t$ is defined as

$$\boldsymbol{A}_t^{\phi} = \frac{\mathrm{d}\mathbf{\Theta}_t^{\mathbf{u}^*+\varepsilon\boldsymbol{\phi}}}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0} \tag{55}$$

and satisfies:

$$\mathrm{d}\boldsymbol{A}_t^{\phi} = \phi_t(\mathbf{\Theta}_t^{\mathbf{u}^*})\,\mathrm{d}t + (\nabla \mathbf{u}^*)^{\top}(\mathbf{\Theta}_t^{\mathbf{u}^*})\boldsymbol{A}_t^{\phi}\,\mathrm{d}t, \qquad \boldsymbol{A}_0^{\phi} = 0. \tag{56}$$

Similarly via the chain rule it follows that:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}_1(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\bigg|_{\varepsilon=0} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\frac{1}{\sqrt{\gamma}}\int_0^1 \mathbf{u}_t^*(\mathbf{\Theta}_t^{\mathbf{u}^*+\varepsilon\boldsymbol{\phi}})^{\top}\mathrm{d}\mathbf{B}_t\right)\bigg|_{\varepsilon=0} = \frac{1}{\sqrt{\gamma}}\int_0^1 \boldsymbol{A}_t^{\phi} \cdot (\nabla \mathbf{u}_t^*)(\mathbf{\Theta}_t^{\mathbf{u}^*})\mathrm{d}\mathbf{B}_t \tag{57}$$

almost surely, combining these results we can see that $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{F}(\mathbf{u}^* + \varepsilon\boldsymbol{\phi})\Big|_{\varepsilon=0} = 0$ almost surely as required.

■

Appendix F. Experimental Details and Further Results

F.1. Method Hyperparameters

In Table F.1 we show the experimental configuration of the trialled algorithms across all datasets.

Method	Hyperparameters	Experiments			
		Step Function	MNIST	LogReg	ICA
N-SFS	Optimser	Adam	Adam	Adam	Adam
	Optimiser step size	10^{-4}	10^{-5}	10^{-4}	10^{-4}
	$\mathbf{\Theta}$ batch size	32	32	32	32
	Data batch size	32	50	Whole train set	10
	# of iterations	300	18750	300	2832
	# of posterior samples	100	100	100	100
	γ	0.05^2	0.1^2	0.2^2	0.01^2
	EM train Δt_{train}	0.05	0.05	0.05	0.05
	EM test Δt_{test}	0.01	0.01	0.01	0.01
SGLD	Adaptive step schedule	$\lambda(i) = \frac{a}{(i+b)^\gamma}$	$\lambda(i) = \frac{a}{(i+b)^\gamma}$	$\lambda(i) = \frac{a}{(i+b)^\gamma}$	$\lambda(i) = \frac{a}{(i+b)^\gamma}$
	a	10^{-3}	10^{-4}	10^{-4}	10^{-4}
	b	10	1	1	1
	γ	0.55	0.55	0.55	0.55
	Posterior Samples	100	100	100	100
	Data batch size	32	32	32	32
	# of iterations	300	18750	300	2832
SGLD	step size	10^{-2}	10^{-3}	-	-
	Data batch size	32	32	-	-
	# of iterations	300	18750	-	-

F.2. Step Function Dataset

Here we describe in detail how the step function dataset was generated:

$$y(x) = \mathbb{1}_{x \geq 0} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.1) \tag{58}$$

Where:

- $\sigma_y = 0.1$
- $N_{\text{train}} = 100$, $N_{\text{test}} = 100$
- $x_{\text{train}} \in (-3.5, 3.5)$
- $x_{\text{test}} \in (-10, 10)$

F.3. Föllmer Drift Architecture

Across all experiments we used the same architecture to parametrise the Föllmer drift:

```python
class SimpleForwardNetBN(torch.nn.Module):

    def __init__(self, input_dim=1, width=20):
        super(SimpleForwardNetBN, self).__init__()

        self.input_dim = input_dim

        self.nn = torch.nn.Sequential(
            torch.nn.Linear(input_dim + 1, width),
            torch.nn.BatchNorm1d(width, affine=False),
            torch.nn.Softplus(),
            torch.nn.Linear(width, width),
            torch.nn.BatchNorm1d(width, affine=False),
            torch.nn.Softplus(),
            torch.nn.Linear(width, width),
            torch.nn.BatchNorm1d(width, affine=False),
            torch.nn.Softplus(),
            torch.nn.Linear(width, width),
            torch.nn.BatchNorm1d(width, affine=False),
            torch.nn.Softplus(),
            torch.nn.Linear(width, input_dim)
        )

        self.nn[-1].weight.data.fill_(0.0)
        self.nn[-1].bias.data.fill_(0.0)
```

Listing 1: Simple architecture for drift.

Note the weights and biases of the final layer are initialised to 0 in order to start the process at a Brownian motion matching the SBP prior.

F.4. BNN Architectures

For the step function dataset we used the following architecture:

```
class DNN_StepFunction(torch.nn.Module):

    def __init__(self, input_dim=1, output_dim=1):
        super(DNN, self).__init__()

        self.output_dim = output_dim
        self.input_dim = input_dim

        self.nn = torch.nn.Sequential(
            torch.nn.Linear(input_dim, 100),
            torch.nn.ReLU(),
            torch.nn.Linear(100, 100),
            torch.nn.ReLU(),
            torch.nn.Linear(100, output_dim)
        )
```

Listing 2: Architecture for step function dataset.

For LeNet5 the exact architecture used was:

```
class LeNet5(torch.nn.Module):

    def __init__(self, n_classes):
        super(LeNet5, self).__init__()

        self.feature_extractor = torch.nn.Sequential(
            torch.nn.Conv2d(
                in_channels=1, out_channels=6,
                kernel_size=5, stride=1
            ),
            torch.nn.Tanh(),
            torch.nn.AvgPool2d(kernel_size=2),
            torch.nn.Conv2d(
                in_channels=6, out_channels=16,
                kernel_size=5, stride=1
            ),
            torch.nn.Tanh(),
            torch.nn.AvgPool2d(kernel_size=2),
        )

        self.classifier = torch.nn.Sequential(
            torch.nn.Linear(in_features=256, out_features=120),
            torch.nn.Tanh(),
            torch.nn.Linear(in_features=120, out_features=84),
            torch.nn.Tanh(),
            torch.nn.Linear(in_features=84, out_features=n_classes),
        )
```

Listing 3: Architecture for MNIST.

Model	Hyperparameters	Values
Step Function	Prior	$\mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbb{I})$
	Likelihood	$\mathcal{N}(\boldsymbol{y}_i \vert f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \sigma_y^2 \mathbb{I})$
	σ_θ	0.3
	σ_y	0.1
MNIST	Prior	$\mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbb{I})$
	Likeihood	$\mathrm{Cat}(f_\theta(\boldsymbol{x}_i))$
	σ_θ	0.3
Log Reg	Prior	$\mathrm{Laplace}(\mathbf{0}, \sigma_\theta,)$
	Likelihood	$\mathrm{Bern}(\mathrm{Sigmoid}_\theta)$
	σ_θ	1
ICA	Prior	$\mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbb{I})$
	Lieklihood	$\prod_i \frac{1}{4\cosh^2(\frac{\boldsymbol{\theta}_i^\top \boldsymbol{x}}{2})}$
	σ_θ	1

Table 5: Specification of Bayesian models.

F.5. Likelihood and Prior Hyperparameters

In Table F.5 we describe the hyperparameters of each Bayesian model as well as their priors and likelihood.