5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–18

Neural Score Estimation: Likelihood-Free Inference with Conditional Score Based Diffusion Models

Anonymous Authors
Anonymous Institution

Abstract

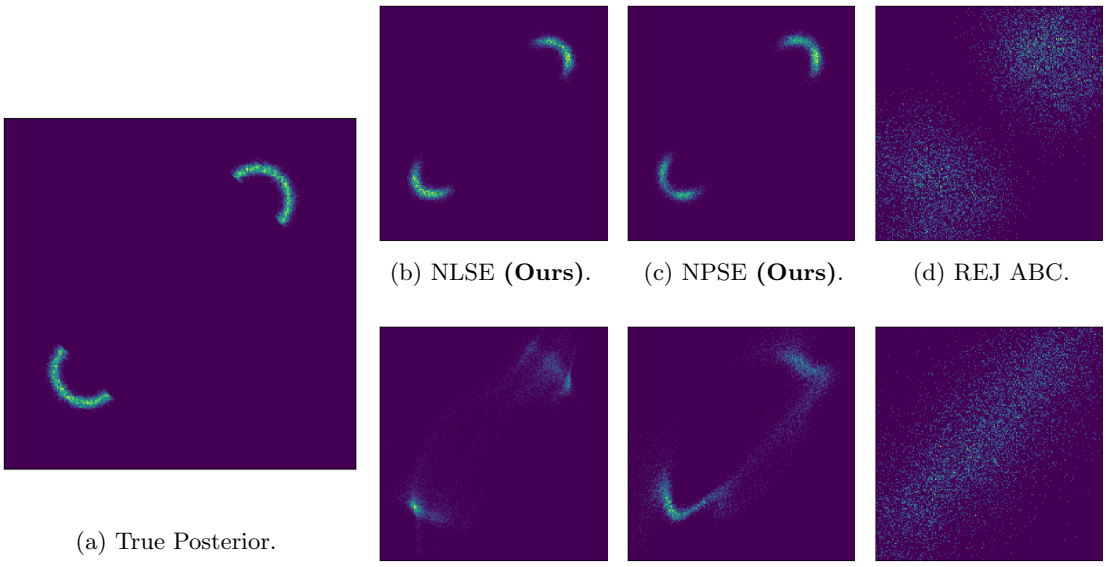
In recent years, score based diffusion models have achieved remarkable empirical performance across a wide range of generative modelling tasks. In this paper, we study the use of conditional score-based diffusion models for Bayesian inference in simulator-based models. We consider two objectives for training these models, one of which approximates the score of the diffused likelihood, while the other directly estimates the score of the diffused posterior. We validate these methods, which we term Neural Posterior Score Estimation (NPSE) and Neural Likelihood Score Estimation (NLSE), on several numerical examples, demonstrating comparable or superior performance to existing state-of-the-art methods such as Neural Posterior Estimation (NPE) and Neural Likelihood Estimation (NLE).

1. Introduction

Many applications in science, engineering, and economics make use of stochastic numerical simulations to model complex phenomena of interest. Such simulator-based models are typically designed by domain experts, using knowledge of the underlying principles of the process of interest. They are thus particularly well suited to domains in which observations are best understood as the result of mechanistic physical processes. These include, amongst others, neuroscience (Gonçalves et al., 2020), evolutionary biology (Beaumont et al., 2002; Ratmann et al., 2007), ecology (Beaumont, 2010; Wood, 2010), epidemiology (Corander et al., 2017), climate science (Holden et al., 2018), cosmology (Alsing et al., 2018), and high-energy physics (Brehmer, 2021).

In many cases, simulator-based models depend on parameters θ which cannot be identified experimentally, and must be inferred from data x. Bayesian inference provides a principled approach for this task. In particular, given a prior $p(\theta)$ and a likelihood $p(x|\theta)$, Bayes' Theorem gives the posterior distribution over the parameters as $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$, where $p(x) = \int_{\mathbb{R}^d} p(x|\theta)p(\theta)\mathrm{d}\theta$ is known as the evidence, or the marginal likelihood. The major difficulty associated with simulator-based models is the absence of a tractable likelihood function $p(x|\theta)$. This inference problem is often referred to as likelihood-free inference or simulation-based inference (SBI) (Cranmer et al., 2020; Sisson et al., 2018).

Traditional methods for performing SBI include approximate Bayesian computation (ABC) (Beaumont et al., 2002; Sisson et al., 2018), whose variants include rejection ABC (Tavaré et al., 1997; Pritchard et al., 1999), MCMC ABC (Marjoram et al., 2003), and sequential Monte Carlo ABC (Beaumont et al., 2009; Bonassi and West, 2015). In such methods, one repeatedly samples parameters, and only accepts parameters for which the samples from the simulator are similar to the observed data x_{obs}.



(a) True Posterior. (b) NLSE **(Ours)**. (c) NPSE **(Ours)**. (d) REJ ABC. (e) NPE. (f) NLE. (g) NRE.

Figure 1: Plots of the posterior samples for the 'two moons' experiment (see Section 3). We plot samples from the true posterior (left), the samples generated by our methods, and the samples generated by several existing SBI methods. We train each algorithm with 1000 samples $(\theta, x) \sim p(\theta)p(x|\theta)$.

In recent years, a range of new SBI methods have been introduced, leveraging advances in machine learning such as normalising flows (Papamakarios et al., 2017, 2021) and generative adversarial networks (Goodfellow et al., 2014). Such methods include Neural Posterior Estimation (NPE) (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), Neural Likelihood Estimation (NLE) (Papamakarios et al., 2019), and Neural Ratio Estimation (NRE) (Durkan et al., 2020; Hermans et al., 2020; Miller et al., 2021; Thomas et al., 2022). Another more recent algorithm, specialising in high-dimensional settings, is Generative Adversarial Training for SBI (Ramesh et al., 2022).

In this paper, inspired by the remarkable success of score-based generative models (Song and Ermon, 2019; Song et al., 2021b; Ho et al., 2020), we consider the application of conditional score-based diffusion models to likelihood-free inference. While such models have previously found success across a wide range of generative modelling tasks (e.g., Batzolis et al., 2021; Dhariwal and Nichol, 2021; Song et al., 2021b; Tashiro et al., 2021), their application to problems of interest to the SBI community (e.g., Lueckmann et al., 2021) has not yet been widely investigated. We note that, in parallel with this work, Geffner et al. (2023) have also studied the use of conditional score-based diffusion models for likelihood-free inference. We provide a more detailed comparison with this paper in Appendix C.

In contrast to existing SBI approaches based on normalising flows (e.g., NLE, NPE), this approach only requires estimates for the gradient of the log density, or score function, of the intractable likelihood or the posterior, which can be estimated using a neural network via score matching (Hyvärinen, 2005; Vincent, 2011; Song et al., 2020). Since we do not require a normalisable model, we avoid the need for any strong restrictions on the model architecture.

In addition, unlike methods based on generative adversarial networks (e.g., Ramesh et al. (2022)), we do not require adversarial training objectives, which are notoriously unstable (Metz et al., 2017; Salimans et al., 2016).

We discuss two training objectives for the conditional diffusion model. The first targets the score of the intractable likelihood, while the second targets the score of the posterior. We refer to these two methods as Neural Likelihood Score Estimation (NLSE) and Neural Posterior Score Estimation (NPSE). We focus solely on the amortised setting, in which we learn a single model to get approximate samples from the posterior $p(\theta|x)$ for any observation $x \in \mathbb{R}^p$. We validate the performance of our methods on several benchmark SBI problems, obtaining comparable or superior performance to other state-of-the-art methods.

2. Likelihood Free Inference with Score-Based Diffusion Models

2.1. Likelihood Free Inference

We focus on the following problem. Suppose a simulator generates $x \in \mathbb{R}^p$ from parameters $\theta \in \mathbb{R}^d$. We assume that the parameters are distributed according to some known prior $p(\theta)$, but that the likelihood $p(x|\theta)$ is intractable. Given an observation x_{obs}, we are interested in sampling from the posterior distribution $p(\theta|x_{\text{obs}}) \propto p(x_{\text{obs}}|\theta)p(\theta)$, given a finite number of i.i.d. samples $(\theta_i, x_i)_{i=1}^n \sim p(\theta)p(x|\theta)$.

2.2. Score-Based Diffusion Models for Likelihood Free Inference

We can tackle this problem using conditional score-based diffusion models (e.g., Song et al., 2021b). In such models, noise is gradually added to the target distribution using a diffusion process, resulting in a tractable reference distribution. The time-reversal of this process is also a diffusion process, whose dynamics we can learn using denoising score matching. We can thus generate samples from the target by simulating the approximate reverse-time process, initialised at samples from the reference distribution.

More precisely, we begin by defining a forward noising process $(\theta_t)_{t\in[0,T]}$ according to

$$\mathrm{d}\theta_t = f(\theta_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t, \quad (\theta_0, x) \sim p(\theta, x), \tag{1}$$

where $f : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ is the drift coefficient, $g : \mathbb{R}_+ \to \mathbb{R}$ is the diffusion coefficient, $(w_t)_{t\geq 0}$ is a standard $\mathbb{R}^d$-valued Brownian motion. We assume that f and g are chosen such that (1) admits a unique stationary distribution p_{ref}, from which it is easy to sample. Under mild conditions, the time-reversed process $(\theta_\tau)_{\tau\in[T,0]} := (\theta_{T-t})_{t\in[0,T]}$, conditioned on x, is also a diffusion process, which satisfies (Anderson, 1982; Föllmer, 1985; Haussmann and Pardoux, 1986; Song et al., 2021b)

$$\mathrm{d}\theta_\tau = \left[f(\theta_\tau, \tau) - g^2(\tau)\nabla_{\theta_\tau} \log p_\tau(\theta_\tau|x)\right] \mathrm{d}\tau + g(\tau)\mathrm{d}w_\tau, \tag{2}$$

where $p_t(\cdot|x)$ denotes the conditional density of θ_t given x. We will suppose that T is sufficiently large such that $p_T \approx p_{\text{ref}}$. Then, by sampling $\theta_T \sim p_{\text{ref}}(\theta)$, and simulating (2), we can obtain samples from the posterior distribution $\theta_0 \sim p_0(\theta|x) := p(\theta|x)$.

In practice, we do not have access to the perturbed posterior scores $\nabla_{\theta_t} \log p_t(\theta_t|x)$, and thus we cannot simulate (2) directly. However, we can obtain an estimate of these scores via (denoising) score matching (e.g., Song et al., 2021b). We will consider two possible approaches to this task.

2.2.1. Neural Posterior Score Estimation

The first approach is based on training a time-varying score network $s_{\psi_{\text{post}}}(\theta_t, x, t) \approx \nabla_{\theta_t} \log p_t(\theta_t|x)$ to directly approximate the score of the perturbed posterior (Batzolis et al., 2021; Dhariwal and Nichol, 2021; Song et al., 2021b). To do so, we would like to minimise the weighted Fisher divergence

$$\frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_t(\theta_t,x)}\left[||s_{\psi_{\text{post}}}(\theta_t, x, t) - \nabla_{\theta_t} \log p(\theta_t|x)||^2\right] \mathrm{d}t, \tag{3}$$

where $\lambda_t : [0, T] \to \mathbb{R}_+$ is a positive weighting function. The second term in this expression is intractable, and thus we cannot minimise it directly. However, one can show that it is equivalent (see Appendix B.1) to minimise the conditional denoising posterior score matching objective (Batzolis et al., 2021; Tashiro et al., 2021)

$$\mathcal{J}_{\text{post}}(\psi_{\text{post}}) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(\theta_0,x)}\left[||s_{\psi_{\text{post}}}(\theta_t, x, t) - \nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)||^2\right] \mathrm{d}t, \tag{4}$$

where $p_{t|0}(\theta_t|\theta_0)$ denotes the transition kernel from θ_0 to θ_t. The expectation in (4) only depends on samples $\theta_0 \sim p(\theta)$ from the prior, $x \sim p(x|\theta)$ from the simulator, and $\theta_t \sim p_{t|0}(\theta|\theta_0)$, from the forward diffusion (1). Moreover, given a suitable choice for the dynamics in (1), $\nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)$ can be computed analytically. We can thus compute a Monte Carlo estimate of (4), and minimise this using, e.g., stochastic gradient descent (SGD).

2.2.2. Neural Likelihood Score Estimation

The second approach is based on the following decomposition of the posterior score,

$$\nabla_{\theta_t} \log p_t(\theta_t|x) = \nabla_{\theta_t} \log p_t(x|\theta_t) + \nabla_{\theta_t} \log p_t(\theta_t). \tag{5}$$

This decomposition suggests that, rather than directly targeting the score of the posterior, we could instead train a score network $s_{\psi_{\text{lik}}}(\theta_t, x, t) \approx \nabla_{\theta_t} \log p_t(x|\theta_t)$ for the likelihood, and then estimate the posterior score using[1]

$$s_{\psi_{\text{post}}}(\theta_t, x, t) = s_{\psi_{\text{lik}}}(\theta_t, x, t) + \nabla_{\theta_t} \log p_t(\theta_t). \tag{6}$$

In order to learn $s_{\psi_{\text{lik}}}(\theta_t, x, t)$, we would like to minimise the weighted Fisher divergence

$$\frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_t(\theta_t,x)}\left[\left|\left|s_{\psi_{\text{lik}}}(\theta_t, x, t) - \nabla_{\theta_t} \log p_t(x|\theta_t)\right|\right|^2\right] \mathrm{d}t. \tag{7}$$

Similar to (3), we cannot optimise this objective due to the intractable second term. However, one can show that it is equivalent (see Appendix B.2) to minimise the corresponding denoising score matching objective function, which in this case is given by

$$\mathcal{J}_{\text{lik}}(\psi_{\text{lik}}) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(\theta_0,x)}\left[\left|\left|s_{\psi_{\text{lik}}}(\theta_t, x, t) + \nabla_{\theta_t} \log p_t(\theta_t) - \nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right] \mathrm{d}t. \tag{8}$$

Similar to (4), we can compute Monte Carlo estimates of the expectations in (8), thus minimise this objective using SGD.

1. It is worth noting that one can only compute $\nabla_{\theta_t} \log p_t(\theta_t)$ directly for certain choices of the prior (see Appendix A.1). If this is not possible, one can instead learn an additional score network $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_{\theta_t} \log p_t(\theta_t)$ for the prior (see Appendix A.2), and then substitute this where required.

Algorithm 1: Neural Posterior Score Estimation (NPSE) and Neural Likelihood Score Estimation (NLSE)

Input: Simulator $p(x|\theta)$, prior $p(\theta)$, simulation budget N
for $i = 1 \dots N$ **do** sample $\theta_{0,i} \sim p(\theta)$, $x_i \sim p(x|\theta_{0,i})$, $(\theta_{t,i})_{t\in(0,T]} \sim p_{t|0}(\theta|\theta_{i,0})$.
if 'NPSE' **then**
  Train $s_{\psi_{\text{post}}}(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ by minimising a Monte Carlo estimate of (4)
else if 'NLSE' **then**
  Train $s_{\psi_{\text{lik}}}(\theta_t, x, t) \approx \nabla_\theta \log p_t(x|\theta_t)$ by minimising a Monte Carlo estimate of (8).
  Set $s_{\psi_{\text{post}}}(\theta_t, x, t) := s_{\psi_{\text{lik}}}(\theta_t, x, t) + \nabla_\theta \log p_t(\theta_t)$.
Simulate the backward diffusion (2), initialised at $\theta_T \sim p_{\text{ref}}(\theta)$, with $s_{\psi_{\text{post}}}(\theta_t, x_{\text{obs}}, t)$.

2.2.3. The Algorithm

We now have all of the necessary ingredients to generate samples from $p(\theta|x_{\text{obs}})$.

(i) Draw samples $\theta_0 \sim p(\theta)$ from the prior, $x \sim p(x|\theta_0)$ from the likelihood, and $\theta_t \sim p_{t|0}(\theta_t|\theta_0)$ using the forward diffusion (1).

(ii) Train a score network $s_{\psi_{\text{post}}}(\theta_t, x, t)$ by minimising the posterior denoising score matching objective (4), or the likelihood denoising score matching objective (8).

(iii) Draw samples $\theta_T \sim p_{\text{ref}}(\theta)$ from the reference distribution. Simulate the backward diffusion (2) with $x = x_{\text{obs}}$, substituting $\nabla_{\theta_t} \log p(\theta_t|x_{\text{obs}}) \approx s_{\psi_{\text{post}}}(\theta_t, x_{\text{obs}}, t)$.

In line with the current SBI taxonomy, we will refer to this approach as Neural Posterior Score Estimation (NPSE) or Neural Likelihood Score Estimation (NLSE), depending on which objective function is used to train the score network.

3. Numerical Experiments

In this section, we provide numerical results for four popular SBI benchmarking experiments: Mixture of Gaussians, Two Moons, Gaussian Linear Uniform, and Simple Likelihood Complex Posterior (Lueckmann et al., 2021). In all experiments, our score network is an MLP with 3 fully connected layers, each with 256 neurons and SiLU activation functions. We use Adam (Kingma and Ba, 2015) to train the networks, with a learning rate of 5×10^{-4} and a batch size of 50. We hold back 10% of the data to be used as a validation set for early stopping. Further experimental details are provided in Appendices D.1 and D.2.

The results for all of these experiments, for simulation budgets of 1000, 10000 and 30000, are displayed in Figure 2. In all experiments, we report the classification-based two-sample test (C2ST) score (Lopez-Paz and Oquab, 2017). The C2ST score varies between 0.5 and 1 (lower being better), with a score of 0.5 indicating perfect posterior estimation. For reference, we compare our methods with Rejection ABC (REJ-ABC) (Tavaré et al., 1997), NRE (Hermans et al., 2020), NLE (Papamakarios et al., 2019), and NPE (Papamakarios and Murray, 2016), implemented using the python toolkit `sbibm` (Lueckmann et al., 2021). Our results indicate that, for all of the experiments considered, our methods (NPSE and NLSE) perform at least as well as these existing methods, particularly for small simulation budgets.

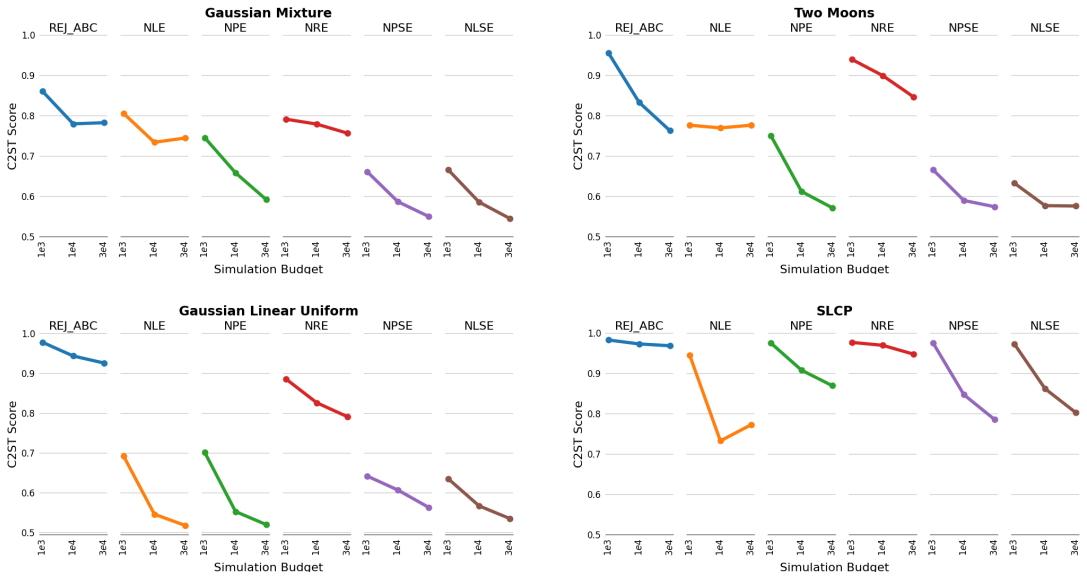


Figure 2: **Performance of NPSE and NLSE on 4 benchmark SBI problems.** Each point represents the mean C2ST score (lower is better), computed over 10 observations.

4. Discussion

A natural question is whether it is preferable to estimate the score of the likelihood or the score of the posterior (see also the related discussions in Greenberg et al. (2019); Papamakarios et al. (2019); Lueckmann et al. (2021)). In general, the best choice of algorithm will depend on the specific problem at hand. Our numerical results indicate that, in many cases, estimating the score of the likelihood (NLSE) can lead to slightly more accurate results than estimating the score of the posterior (NPSE). This being said, it is worth noting that in all our experiments it was possible to compute the score of perturbed prior analytically. When this is not possible, NLSE requires us to approximate this term by fitting an additional score network. In typical SBI problems, it is very cheap to sample from the prior, particularly compared to the simulator. Thus, in principle, one could use as many samples as required to obtain a sufficiently accurate estimate for the prior score. Nonetheless, in such cases, it may be preferable to use NPSE, which only requires us to estimate the posterior score.

In practice, it is not uncommon to be solely interested in generating samples from $p(\theta|x_{\text{obs}})$, i.e., the posterior distribution of the parameters given a specific observation x_{obs}. In this case, at the expense of additional compute, sequential variants of existing methods such as SNPE (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), SNLE (Lueckmann et al., 2019; Papamakarios et al., 2019), and SNRE (Durkan et al., 2020; Hermans et al., 2020; Miller et al., 2021; Thomas et al., 2022) can lead to significant performance improvements over their non-sequential (i.e., amortised) counterparts, by guiding simulations using a sequence of carefully chosen proposal priors. In this context, an interesting direction for future work is to develop an effective sequential version of NPSE and NLSE, along the lines of these existing schemes. We refer to Sharrock et al. (2022) for some ongoing work in this direction.

References

Justin Alsing, Benjamin Wandelt, and Stephen Feeney. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly Notices of the Royal Astronomical Society*, 477(3):2874–2885, jul 2018. ISSN 0035-8711. doi: 10.1093/mnras/sty819.

Brian D O Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90051-5.

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional Image Generation with Score-Based Diffusion Models. *arXiv preprint*, 2021. doi: 10.48550/arXiv.2111.13606.

Mark A Beaumont. Approximate Bayesian Computation in Evolution and Ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41(1):379–406, nov 2010. ISSN 1543-592X. doi: 10.1146/annurev-ecolsys-102209-144621.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035, dec 2002. ISSN 1943-2631. doi: 10.1093/genetics/162.4.2025.

Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, dec 2009. ISSN 0006-3444. doi: 10.1093/biomet/asp052.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

Fernando V Bonassi and Mike West. Sequential Monte Carlo with Adaptive Weights for Approximate Bayesian Computation. *Bayesian Analysis*, 10(1):171–187, mar 2015. doi: 10.1214/14-BA891.

Johann Brehmer. Simulation-based inference in particle physics. *Nature Reviews Physics*, 3(5):305, 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00305-6.

Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. Denoising likelihood score matching for conditional score-based data generatio. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022.

Jukka Corander, Christophe Fraser, Michael U Gutmann, Brian Arnold, William P Hanage, Stephen D Bentley, Marc Lipsitch, and Nicholas J Croucher. Frequency-dependent selection in vaccine-associated pneumococcal population dynamics. *Nature Ecology & Evolution*, 1(12):1950–1960, 2017. ISSN 2397-334X. doi: 10.1038/s41559-017-0337-x.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, dec 2020. doi: 10.1073/pnas.1912789117.

Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021.

Conor Durkan, Iain Murray, and George Papamakarios. On Contrastive Learning for Likelihood-free Inference. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online, 2020.

H Föllmer. An entropy approach to the time reversal of diffusion processes. In M Metivier and E Pardoux, editors, *Stochastic Differential Systems Filtering and Control*, pages 156–163. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985. ISBN 978-3-540-39253-8.

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional Score Modeling for Simulation-based Inference. In *Proceedings of the 40th International Conference of Machine Learning (ICML 2023)*, Honolulu, HI, 2023.

Manuel Glockler, Michael Deistler, and Jakob H Macke. Variational Methods for Simulation-Based Inference. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022.

Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, David S Greenberg, and Jakob H Macke. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261, 2020. ISSN 2050-084X. doi: 10.7554/eLife.56261.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS 2014)*, pages 2672–2680, Montreal, Canada, 2014.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatics Posterior Transformation for Likelihood-Free Inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, 2019.

U G Haussmann and E Pardoux. Time Reversal of Diffusions. *The Annals of Probability*, 14(4):1188–1205, oct 1986. doi: 10.1214/aop/1176992362.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, Online, 2020.

Philip B. Holden, Neil R. Edwards, James Hensman, and Richard D. Wilkinson. ABC for Climate: Dealing with Expensive Simulators. In S. A. Sisson, Y. Fan, and M. A. Beaumont, editors, *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, New York, 2018. doi: 10.1201/9781315117195.

Aapo Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimisation. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR '15)*, pages 1–13, San Diego, CA, 2015.

David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, 2022.

Jan-Matthis Lueckmann, Pedro J. Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H. Macke. Flexible Statistical Inference for Mechanistic Models of Neural Dynamics. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, 2017.

Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H Macke. Likelihood-free inference with emulator networks. In *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, volume 96, pages 32–53, 2019.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking Simulation-Based Inference. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021)*, pages 343–351, Online, 2021.

Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, dec 2003. doi: 10.1073/pnas.0306899100.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled Generative Adversarial Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017.

Benjamin Kurt Miller, Alex Cole, Patrick Forré, Gilles Louppe, and Christoph Weniger. Truncated Marginal Neural Ratio Estimation. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021.

George Papamakarios and Iain Murray. Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation. In *Proceedings of the 30th Conference on Neural Information Processings Systems (NIPS 2016)*, Barcelona, Spain, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked Autoregressive Flow for Density Estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, NIPS'17, pages 2335–2344, Red Hook, NY, 2017. Curran Associates Inc. ISBN 9781510860964.

George Papamakarios, David C. Sterratt, and Iain Murray. Sequential Neural Likelihood: Fast Likelihood-Free Inference with Autoregressive Flows. In *22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Okinawa, Japan, 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

J K Pritchard, M T Seielstad, A Perez-Lezaun, and M W Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, dec 1999. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a026091.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. GATSBI: Generative Adversarial Training for Simulation-Based Inference. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022.

Oliver Ratmann, Ole Jørgensen, Trevor Hinkley, Michael Stumpf, Sylvia Richardson, and Carsten Wiuf. Using Likelihood-Free Inference to Compare Evolutionary Dynamics of the Protein Networks of H. pylori and P. falciparum. *PLOS Computational Biology*, 3 (11):e230, nov 2007. doi: 10.1371/journal.pcbi.0030230.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved Techniques for Training GANs. In *Proceedings of the 30th Conference on Neural Information Processings Systems (NIPS 2016)*, volume 29, Barcelona, Spain, 2016.

Louis Sharrock, Jack Simons, Song Liu, and Mark Beaumont. Sequential Neural Score Estimation: Likelihood-Free Inference with Conditional Score Based Diffusion Models. *arXiv preprint*, 2022. doi: 10.48550/arXiv.2210.04872.

S A Sisson, Y Fan, and Mark M Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, feb 2007. doi: 10.1073/pnas.0607208104.

S.A. Sisson, Y. Fan, and M. A. Beaumont. Overview of Approximate Bayesian Computation. In *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC Press., New York, 2018. doi: 10.1201/9781315117195.

Y. Song, C. Durkan, I. Murray, and S. Ermon. Maximum Likelihood Training of Score-Based Diffusion Models. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021a.

Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, Online, 2021b.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019.

Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, Online, 2020.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced Score Matching: A Scalable Approach to Density and Score Estimation. In *Uncertainty in Artificial Intelligence*, Online, 2020.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional Score-Based Diffusion Models for Probabilistic Time Series Imputation. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021.

Simon Tavaré, David J Balding, R C Griffiths, and Peter Donnelly. Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145(2):505–518, feb 1997. ISSN 1943-2631. doi: 10.1093/genetics/145.2.505.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-Free Inference by Ratio Estimation. *Bayesian Analysis*, 17(1):1–31, mar 2022. doi: 10.1214/20-BA1238.

P Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Samuel Wiqvist, Jes Frellsen, and Umberto Picchini. Sequential Neural Posterior and Likelihood Approximation. *arXiv preprint*, 2021. doi: 10.48550/arXiv.2102.06522.

Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010. ISSN 1476-4687. doi: 10.1038/nature09319.

Qinsheng Zhang and Yongxin Chen. Fast Sampling of Diffusion Models with Exponential Integrator. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, Kigali, Rwanda, 2023.

Appendix A. Computing or Estimating the Perturbed Prior Score

NLSE may require us to compute or to estimate the score of the perturbed prior. In this appendix, we provide further details on how to do so.

A.1. Computing the Perturbed Prior Score

For certain choices of the prior, we can obtain the perturbed prior $p_t(\theta_t) = \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta)p(\theta)\mathrm{d}\theta$ in closed form. We can then obtain the score of the perturbed prior $\nabla_{\theta_t} \log p_t(\theta_t)$ using automatic differentiation. We provide two examples below.

A.1.1. Uniform Prior

Suppose $p(\theta) = \mathcal{U}(\theta|a, b)$ and $p_{t|0}(\theta_t|\theta) = \mathcal{N}(\theta_t|\theta, \tau_t^2 I)$. We then have

$$\begin{aligned} p_t(\theta_t) &= \int_{\mathbb{R}^d} p(\theta)p_{t|0}(\theta_t|\theta)\mathrm{d}\theta \\ &= \frac{1}{\prod_{i=1}^d (b_i - a_i)} \int_{[a_1,b_1]\times\cdots\times[a_d,b_d]} \mathcal{N}(\theta_t|\theta, \tau_t^2 I)\mathrm{d}\theta \\ &= \frac{1}{\prod_{i=1}^d (b_i - a_i)} \prod_{i=1}^d \left(\Phi(b_i|\theta_{t,i}, \tau_{i,t}^2) - \Phi(a_i|\theta_{t,i}, \tau_{i,t}^2)\right), \end{aligned}$$

where $\Phi(\cdot|\mu, \sigma^2)$ is the CDF of a univariate Gaussian with mean μ and variance σ^2.

A.1.2. Gaussian Mixture Prior

Suppose that $p(\theta) = \sum_{i=1}^n \alpha_i \mathcal{N}(\theta|\mu_i, \Sigma_i)$ and $p_{t|0}(\theta_t|\theta) = \mathcal{N}(\theta_t|\theta, \tau_t^2 I)$. Using standard results (e.g., Equation 2.115 in Bishop, 2006), it follows straightforwardly that

$$\begin{aligned} p_t(\theta_t) &= \int_{\mathbb{R}^d} p(\theta)p_{t|0}(\theta_t|\theta)\mathrm{d}\theta \\ &= \sum_{i=1}^n \alpha_i \int_{\mathbb{R}^d} \mathcal{N}(\theta|\mu_i, \Sigma_i)\mathcal{N}(\theta_t|\theta, \tau_t^2 I)\mathrm{d}\theta = \sum_{i=1}^n \alpha_i \mathcal{N}\left(\theta_t|\mu_i, \Sigma_i + \tau_t^2 I\right). \end{aligned}$$

A.2. Estimating the Perturbed Prior Score

In cases where it is not possible to obtain the perturbed prior in closed form (e.g., if the prior is implicit), we can instead learn an approximation $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_{\theta_t} \log p_t(\theta_t)$ by minimising the standard score matching objective

$$\frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{\theta_t \sim p_t(\theta_t)} \left[||s_{\psi_{\text{pri}}}(\theta_t, t) - \nabla_{\theta_t} \log p_t(\theta_t)||^2\right] \mathrm{d}t. \tag{9}$$

This objective is, of course, intractable. However, as is now well known, it is equivalent to minimise the denoising objective (e.g., Song et al., 2021b)

$$\mathcal{J}_{\text{pri}}(\psi_{\text{pri}}) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{\theta_0 \sim p(\theta_0)\theta_t \sim p_{t|0}(\theta_t|\theta_0)} \left[||s_{\psi_{\text{pri}}}(\theta_t, t) - \nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)||^2\right] \mathrm{d}t. \tag{10}$$

Algorithm 2: Prior Score Estimation

Input: Prior $p(\theta)$, simulation budget M
for $i = 1 \dots M$ **do** sample $\theta_{0,i} \sim p(\theta)$, $(\theta_{t,i})_{t\in(0,T]} \sim p_{t|0}(\theta|\theta_{i,0})$.
Train $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_\theta \log p_t(\theta_t)$ by minimising a Monte Carlo estimate of (10)

Appendix B. Derivations for Neural Posterior Score Estimation (NPSE) and Neural Likelihood Score Estimation (NLSE)

In this appendix, we derive the denoising objective functions used in NPSE and NLSE. We provide these derivations for completeness, noting that similar results can also be found in Batzolis et al. (2021) and Chao et al. (2022), respectively.

B.1. Neural Posterior Score Estimation (NPSE)

In NPSE, the posterior score matching objective function is given by

$$\begin{aligned}
\mathcal{J}_{\text{post}}(\psi_{\text{post}}) &= \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||s_{\psi_{\text{post}}}(\theta_t,x,t) - \nabla_{\theta_t}\log p_t(\theta_t|x)||^2\right]\mathrm{d}t \\
&= \frac{1}{2}\int_0^T \lambda_t \Bigg[\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\right]}_{\Omega_t^1} \\
&\qquad - 2\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}[s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_t(\theta_t|x)]}_{\Omega_t^2} \\
&\qquad + \underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||\nabla_{\theta_t}\log p_t(\theta_t|x)||^2\right]}_{\Omega_t^3}\Bigg]\mathrm{d}t.
\end{aligned}$$

For the first term Ω_t^1, we have that

$$\begin{aligned}
\Omega_t^1 &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x && \text{(definition of } \mathbb{E}) \\
&= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t|x)p(x)||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x && \text{(Bayes' Theorem for } p_t(\theta_t,x)) \\
&= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\left[\int_{\mathbb{R}^n} p_{t|0}(\theta_t|x,\theta_0)p(\theta_0|x)\mathrm{d}\theta_0\right]p(x)||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x \\
& && \text{(law of total probability)} \\
&= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p_{t|0}(\theta_t|\theta_0)p(\theta_0|x)p(x)||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x \\
& && \text{(conditional independence of } \theta_t, x|\theta_0) \\
&= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2\right]. && \text{(definition of } \mathbb{E})
\end{aligned}$$

For the second term Ω_t^2, we have that

$$\Omega_t^2 = \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \qquad \text{(definition of } \mathbb{E})$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t|x)p(x)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \quad \text{(Bayes' Theorem for } p_t(\theta_t,x))$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p(x)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \qquad (\nabla_{\theta_t}\log p_t(\theta_t|x) = \tfrac{\nabla_{\theta_t}p_t(\theta_t|x)}{p_t(\theta_t|x)})$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p(x)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\left[\int p_{t|0}(\theta_t|x,\theta_0)p(\theta_0|x)\mathrm{d}\theta_0\right]\mathrm{d}\theta_t\mathrm{d}x$$

(law of total probability)

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(\theta_0,x)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x$$

(conditional independence of $\theta_t, x|\theta_0$)

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(\theta_0,x)p_{t|0}(\theta_t|\theta_0)s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x$$

$$(\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0) = \tfrac{\nabla_{\theta_t}p_{t|0}(\theta_t|\theta_0)}{p_{t|0}(\theta_t|\theta_0)})$$

$$= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}[s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)]. \qquad \text{(definition of } \mathbb{E})$$

The third term Ω^3_t is independent of ψ_{post}. We thus have

$$\begin{aligned}\mathcal{J}_{\text{post}}(\psi_{\text{post}}) &\propto \frac{1}{2}\int_0^t \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\Big[||s_{\psi_{\text{post}}}(\theta_t,x,t)||^2 \\ &\qquad -2s^T_{\psi_{\text{post}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\Big]\,\mathrm{d}t \\ &\propto \frac{1}{2}\int_0^t \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_{\psi_{\text{post}}}(\theta_t,x,t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)||^2\right]\mathrm{d}t.\end{aligned}$$

B.2. Neural Likelihood Score Estimation (NLSE)

In NLSE likelihood score matching objective function is given by

$$\begin{aligned}\mathcal{J}_{\text{lik}}(\psi_{\text{lik}}) &= \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||s_{\psi_{\text{lik}}}(\theta_t,x,t) - \nabla_{\theta_t}\log p_t(x|\theta_t)||^2\right]\mathrm{d}t \\ &= \frac{1}{2}\int_0^T \lambda_t\Bigg[\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}[||s_{\psi_{\text{lik}}}(\theta_t,x,t)||^2}_{\Omega^1_t} \\ &\qquad - 2\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}[s^T_{\psi_{\text{lik}}}(\theta_t,x,t)\nabla_{\theta_t}\log p_t(x|\theta_t)]}_{\Omega^2_t} \\ &\qquad + \underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}[||\nabla_{\theta_t}\log p_t(x|\theta_t)||^2]}_{\Omega^3_t}\Bigg]\mathrm{d}t.\end{aligned}$$

We now proceed in much the same vein as before. On this occasion, leveraging the same arguments as in Appendix B.1, it is straightforward to obtain

$$\Omega^1_t = \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)||s_{\psi_{\text{lik}}}(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x \qquad \text{(definition of } \mathbb{E})$$

$$= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_{\psi_{\text{lik}}}(\theta_t,x,t)||^2\right], \qquad \text{(identically to } \Omega_t^1 \text{ in Appendix B.1)}$$

$$\Omega_t^2 = \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)s_{\psi_{\text{lik}}}^T(\theta_t,x,t)\nabla_{\theta_t}\log p_t(x|\theta_t)\mathrm{d}\theta_t\mathrm{d}x \qquad \text{(definition of } \mathbb{E})$$
$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)s_{\psi_{\text{lik}}}^T(\theta_t,x,t)[\nabla_{\theta_t}\log p_t(\theta_t|x) - \nabla_{\theta_t}\log p_t(\theta_t)]\mathrm{d}\theta_t\mathrm{d}x$$
(Bayes' Theorem)

$$= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta\sim p_{t|0}(\theta_t|\theta_0)}\left[s_{\psi_{\text{lik}}}^T(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0) - s_{\psi_{\text{lik}}}^T(\theta_t,x,t)\nabla_{\theta_t}\log p_t(\theta_t)\right],$$
(from Appendix B.1)

and, of course, that Ω_t^3 is independent of ψ_{lik}. Putting everything together, we thus have

$$\mathcal{J}_{\text{lik}}(\psi_{\text{lik}}) \propto \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta\sim p_{t|0}(\theta_t|\theta_0)}\big[||s_{\psi_{\text{lik}}}(\theta_t,x,t)||^2 - 2s_{\psi_{\text{lik}}}^T(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0) + 2s_{\psi_{\text{lik}}}^T(\theta_t,x,t)\nabla_{\theta_t}\log p_t(\theta_t)\big]\mathrm{d}t$$

$$\propto \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta\sim p_{t|0}(\theta_t|\theta_0)}\big[||s_{\psi_{\text{lik}}}(\theta_t,x,t) - \big(\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0) - \nabla_{\theta_t}\log p_t(\theta_t)\big)||^2\big]\mathrm{d}t.$$

Appendix C. Likelihood-Free Inference with Multiple Observations

In this appendix, we discuss how to adapt NPSE and NLSE to the task of generating samples from $p(\theta_t|x_{\text{obs}}^1,\dots,x_{\text{obs}}^n)$ for any set of observations $\{x_{\text{obs}}^1,\dots,x_{\text{obs}}^n\}$ (Geffner et al., 2023).

C.1. Neural Posterior Score Estimation

In Geffner et al. (2023), the authors observe that it is not possible to factorise the multiple-observation posterior score $\nabla_\theta \log p_t(\theta|x_{\text{obs}}^1,\dots,x_{\text{obs}}^n)$ in terms of the single-observation posterior scores $\nabla_\theta \log p_t(\theta|x_{\text{obs}}^i)$, and the prior score $\nabla_\theta \log p(\theta)$. Thus, a naive implementation of NPSE would require training a network $s_{\psi_{\text{post}}}(\theta,x^1,\dots,x^n,t) \approx \nabla_\theta \log p_t(\theta|x^1,\dots,x^n)$ using samples $(\theta,x^1,\dots,x^n) \sim p(\theta)\prod_{j=1}^n p(x^i|\theta)$. This requires calling the simulator n times for every parameter sample θ, and is thus highly sample inefficient.

To circumvent this issue, Geffner et al. (2023) introduce a new method based on the observation that $p(\theta|x^1,\dots,x^n) \propto p(\theta)^{1-n}\prod_{i=1}^n p(\theta|x^i)$. In particular, they propose to use the sequence of densities

$$p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) \propto (p(\theta)^{1-n})^{\frac{T-t}{T}}\prod_{i=1}^n p_t(\theta|x^i). \tag{11}$$

Importantly, the density at $t=0$ coincides with the target distribution $p(\theta|x^1,\dots,x^n)$, while the density at $t=T$ is is a tractable Gaussian. In addition, the score of these densities can be decomposed into the single-observation posterior scores $\nabla_\theta \log p_t(\theta|x^i)$, and the (known)

prior score $\nabla_\theta \log p(\theta)$, as

$$\nabla_\theta \log p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) = \frac{(1-n)(T-t)}{T}\nabla_\theta \log p(\theta) + \sum_{i=1}^{n} \nabla_\theta \log p_t(\theta|x^i). \tag{12}$$

Thus, in particular, it is only necessary to learn a single score network $s_{\psi_{\text{post}}}(\theta, x, t) \approx \nabla_\theta \log p_t(\theta|x)$, which can be trained using samples $(\theta, x) \sim p(\theta)p(x|\theta)$. After learning this score network, one can then generate samples from the posterior by running the reverse diffusion with

$$\nabla_{\theta_t} \log p_t^{(\text{bridge})}(\theta|x^1_{\text{obs}},\dots,x^n_{\text{obs}}) \approx \frac{(1-n)(T-t)}{T}\nabla_\theta \log p(\theta) + \sum_{i=1}^{n} s_{\psi_{\text{post}}}(\theta, x^i_{\text{obs}}, t).$$

It is worth emphasising that, other than at time $t = 0$, the sequence of bridging densities $p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n)$ do not coincide with the true perturbed multi-observation posterior densities $p_t(\theta|x^1,\dots,x^n)$. Thus, directly solving the reverse-time SDE using a 'predictor-only' method (e.g., Euler-Maruyama) would not result in samples from the posterior, even if one could perfectly estimate the scores of these densities. Instead, we must use a corrector-only method (e.g., annealed Langevin dynamics as in Algorithm 1, Geffner et al., 2023) or a predictor-corrector method (Song et al., 2021a) to solve the reverse-time SDE.

We now provide details of an alternative approach, based on a very similar idea to the one in Geffner et al. (2023). In particular, in place of (11), one could instead use the sequence of densities

$$p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) \propto (p_t(\theta))^{1-n} \prod_{i=1}^{n} p_t(\theta|x^i). \tag{13}$$

This sequence of densities has all of the desirable properties of (11). Once again, the density at $t = 0$ coincides with the target distribution $p(\theta|x^1,\dots,x^n)$, and the density at $t = T$ corresponds to a tractable Gaussian. Moreover, we can factorise these densities in terms of the single-observation posterior scores $\nabla_\theta \log p_t(\theta|x^i)$, and the perturbed prior score $\nabla_\theta \log p_t(\theta)$, as

$$\nabla_\theta \log p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) = (1-n)\nabla_\theta \log p_t(\theta) + \sum_{i=1}^{n} \nabla_\theta \log p_t(\theta|x^i). \tag{14}$$

Similar to above, it is then only necessary to learn a single score network $s_{\psi_{\text{post}}}(\theta, x, t) \approx \nabla_\theta \log p_t(\theta|x)$, which we can train using samples $(\theta, x) \sim p(\theta)p(x|\theta)$. Clearly, the expressions in (13) - (14) are very similar to the ones given in (11) - (12), with the only difference appearing in the first term. These quantities coincide at time zero, but will otherwise differ. The advantage of (12), i.e., the scheme proposed in Geffner et al. (2023), is that it only requires access to the score of the prior, $\nabla_\theta \log p(\theta)$, and is thus very straightforward to implement.

C.2. Neural Likelihood Score Estimation

Using the sequence of densities introduced above, we can also extend NLSE to the multi-observation setting. Observe that we can rewrite the sequence of densities in (13) as

$$p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) \propto p_t(\theta)\prod_{i=1}^{n} p_t(x^i|\theta).$$

Thus, in particular, we can express the score of these densities in terms of the single-observation likelihood scores $\nabla_\theta \log p_t(x^i|\theta)$ as

$$\nabla_\theta \log p_t^{(\text{bridge})}(\theta|x^1,\dots,x^n) = \nabla_\theta \log p_t(\theta) + \sum_{i=1}^{n} \nabla_\theta \log p_t(x^i|\theta),$$

Once again, following this decomposition, it is clear that we are only required to train a single score network, this time for the likelihood score $s_{\psi_{\text{lik}}}(\theta, x, t) \approx \nabla_\theta \log p_t(x|\theta)$, which we can do using samples $(\theta, x) \sim p(\theta)p(x|\theta)$.

Appendix D. Additional Details for Numerical Experiments

D.1. Experiment details

"Gaussian Mixture". This task, introduced by Sisson et al. (2007), appears frequently in the SBI literature (Beaumont et al., 2009; Lueckmann et al., 2021). It consists of a uniform prior $p(\theta) = \mathcal{U}(-10, 10)$, and a simulator given by $p(x|\theta) = 0.5\mathcal{N}(x|\theta, I) + 0.5\mathcal{N}(x|\theta, 0.01I)$, where $\theta, x \in \mathbb{R}^2$.

"Two Moons". This two-dimensional experiment consists of a uniform prior given by $p(\theta) = \mathcal{U}(-1, 1)$, $\theta \in \mathbb{R}^2$, and a simulator defined by

$$x|\theta = \begin{pmatrix} r\cos(\alpha) + 0.25 \\ r\sin(\alpha) \end{pmatrix} + \begin{pmatrix} -|\theta_1 + \theta_2|/\sqrt{2} \\ (-\theta_1 + \theta_2)/\sqrt{2} \end{pmatrix}, \tag{15}$$

where $\alpha \sim \mathcal{U}(-\pi/2, \pi/2)$ and $r \sim \mathcal{N}(0.1, 0.01^2)$ (Greenberg et al., 2019). It defines a posterior distribution over the parameters which exhibits both local (crescent shaped) and global (bimodal) features, and is frequently used to analyse how SBI algorithms deal with multimodality (Greenberg et al., 2019; Wiqvist et al., 2021; Ramesh et al., 2022; Glockler et al., 2022).

"Gaussian Linear Uniform". This task consists of a uniform prior $p(\theta) = \mathcal{U}(-1, 1)$, and a Gaussian simulator $p(x|\theta) = \mathcal{N}(x|\theta, 0.1I)$, where $\theta, x \in \mathbb{R}^{10}$ (Lueckmann et al., 2021). This example allows us to determine how algorithms scale with increased dimensionality, as well as with truncated support.

"Simple Likelihood, Complex Posterior". This challenging task, introduced by Papamakarios et al. (2019), is designed to have a simple likelihood and a complex posterior. The prior is a five-dimensional uniform distribution $p(\theta) = \mathcal{U}(-3, 3)$, while the likelihood for the eight-dimensional data is Gaussian, but with mean and covariance which are highly non-linear functions of the parameters. This defines a complex posterior distribution over the parameters, with four symmetrical modes and vertical cut-offs. For full details, we refer to Appendix A in Papamakarios et al. (2019), or Appendix T in Lueckmann et al. (2021).

D.2. Algorithmic details

In all of our experiments, we perturb samples using the variance exploding SDE (Song et al., 2021b)

$$\mathrm{d}\theta_t = \sigma_{\min}\left(\frac{\sigma_{\min}}{\sigma_{\max}}\right)^t \sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}\mathrm{d}w_t\ , \quad t \in (0,1]. \tag{16}$$

We set $\sigma_{\min} = 0.01$ for the Gaussian Mixture, Two Moons, and Gaussian Linear Uniform experiments, and $\sigma_{\min} = 0.05$ for SLCP. Meanwhile, $\sigma_{\max}$ is chosen according to Technique 1 in Song and Ermon (2020). The variance exploding SDE defines the transition density

$$p_{t|0}(\theta_t|\theta_0) = \mathcal{N}\left(\theta_t \middle| \theta_0, \sigma_{\min}^2\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}\right), \tag{17}$$

with the corresponding score function given by

$$\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0) = -\frac{(\theta_t - \theta_0)}{\sigma_{\min}^2}\left(\frac{\sigma_{\min}}{\sigma_{\max}}\right)^{2t}. \tag{18}$$

We solve the forward SDE using an Euler-Maruyama (EM) discretisation, defined over equally spaced points $\{t_i\}_{i=1}^N$. This results in a geometric sequence of noise perturbations $\{\sigma_i\}_{i=1}^N$, where $\sigma_i = \sigma_{\min}\left(\sigma_{\max}/\sigma_{\min}\right)^{\frac{i-1}{N-1}}$. In our numerics, we fix $\gamma = \frac{\sigma_i}{\sigma_{i-1}} = 0.6$, the ratio between noise perturbations. This, together with our previous specification of $\sigma_{\min}$ and $\sigma_{\max}$, fully determines the number of noise levels N. We solve the corresponding backward SDE using a corrector only method (i.e., annealed Langevin dynamics); see also Algorithm 1 in Song and Ermon (2019). In all experiments, we use 1000 corrector steps, and set the step size for the final round of annealed Langevin dynamics as $\epsilon = 5 \times 10^{-6}$.

D.3. Sampling Time

One of the well known disadvantages of score-based diffusion models is that sampling is slow compared to other generative models, such as generative adversarial networks (Goodfellow et al., 2014). In comparison to other SBI methods, the sampling time of our methods (NPSE and NLSE) is similar to that of other methods which require MCMC to get samples from the posterior, namely NLE (Lueckmann et al., 2019; Papamakarios et al., 2019) and NRE (Durkan et al., 2020; Hermans et al., 2020; Miller et al., 2021; Thomas et al., 2022); but compares unfavourably to methods which only require a single forward pass, e.g. NPE (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019). This being said, it is worth noting that there has been some recent work to address the slow sampling time of diffusion models (e.g., Lu et al., 2022; Zhang and Chen, 2023), which could be used to alleviate this problem in our setting.