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

Louis Sharrock [*] [1] [2] **Jack Simons** [*] [2] **Song Liu** [2] **Mark Beaumont** [2]

Abstract

We introduce Sequential Neural Posterior Score Estimation (SNPSE), a score-based method for Bayesian inference in simulator-based models. Our method, inspired by the remarkable success of score-based methods in generative modelling, leverages conditional score-based diffusion models to generate samples from the posterior distribution of interest. The model is trained using an objective function which directly estimates the score of the posterior. We embed the model into a sequential training procedure, which guides simulations using the current approximation of the posterior at the observation of interest, thereby reducing the simulation cost. We also introduce several alternative sequential approaches, and discuss their relative merits. We then validate our method, as well as its amortised, non-sequential, variant on several numerical examples, demonstrating comparable or superior performance to existing state-of-the-art methods such as Sequential Neural Posterior Estimation (SNPE).

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 often designed by domain experts, using knowledge of the underlying principles of the process of interest. They are thus well suited to domains in which observations are best understood as the result of mechanistic physical processes. These include, amongst others, neuroscience (Sterratt et al., 2011; 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), high-energy physics (Brehmer, 2021), and econometrics (Gourieroux et al., 1993).

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

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \tag{1}$$

where $p(x) = \int p(x|\theta)p(\theta)\mathrm{d}\theta$ is known as the evidence or marginal likelihood. The major difficulty associated with simulator-based models is the absence of a tractable likelihood function $p(x|\theta)$. This precludes, in particular, the use of conventional likelihood-based Bayesian inference methods such as Markov chain Monte Carlo (MCMC) (Brooks et al., 2011) or variational inference (VI) (Blei et al., 2017). The resulting 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 (SMC) ABC (Beaumont et al., 2009; Bonassi & West, 2015). In such methods, one repeatedly samples parameters, and only accepts parameters for which the corresponding samples from the simulator are similar to the observed data x_{obs}.

More recently, a range of new SBI methods have been introduced, which leverage advances in machine learning such as normalising flows (Papamakarios et al., 2017; 2021) and generative adversarial networks (Goodfellow et al., 2014). These methods often include a sequential training procedure, which adaptively guides simulations to yield more informative data. Such methods include Sequential Neural Posterior Estimation (SNPE) (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), Sequential Neural Likelihood Estimation (SNLE) (Lueckmann

[*] Equal contribution [1] Department of Mathematics and Statistics, Lancaster University, UK [2] School of Mathematics, University of Bristol, UK. Correspondence to: Louis Sharrock <l.sharrock@lancaster.ac.uk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

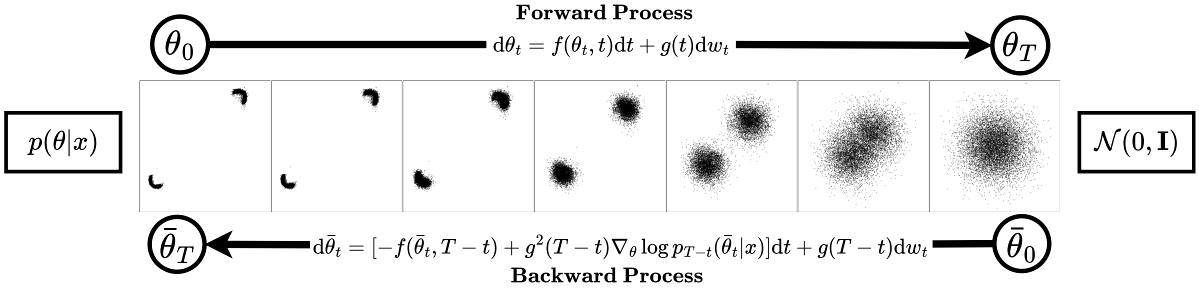

Forward Process
θ_0
$\mathrm{d}\theta_t = f(\theta_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t$
θ_T
$p(\theta|x)$
$\mathcal{N}(0, \mathbf{I})$
$\bar{\theta}_T$
$\mathrm{d}\bar{\theta}_t = [-f(\bar{\theta}_t, T-t) + g^2(T-t)\nabla_\theta \log p_{T-t}(\bar{\theta}_t|x)]\mathrm{d}t + g(T-t)\mathrm{d}w_t$
$\bar{\theta}_0$
Backward Process

Figure 1. **Visualisation of posterior inference using Neural Posterior Score Estimation (NPSE) in the 'Two Moons' experiment.** The forward process transforms samples from the target posterior distribution $p(\theta|x)$ to a tractable reference distribution. The backward process transports samples from the reference to the target posterior. The backward process depends on the scores $\nabla_\theta \log p_t(\theta|x)$, which can be estimated using score matching techniques given access to samples $(\theta, x) \sim p(\theta)p(x|\theta)$ (see Section 2.2).

et al., 2019; Papamakarios et al., 2019), and Sequential Neural Ratio Estimation (SNRE) (Durkan et al., 2020; Hermans et al., 2020; Miller et al., 2021; Thomas et al., 2022). Other more recent algorithms of a similar flavour include Sequential Neural Variational Inference (SNVI) (Glockler et al., 2022), Generative Adversarial Training for SBI (GATSBI) (Ramesh et al., 2022), Truncated SNPE (TSNPE) (Deistler et al., 2022a), and Sequential Unnormalized Neural Likelihood Estimation (SUNLE) (Glaser et al., 2022).

In this paper, we present Neural Posterior Score Estimation (NPSE), as well as its sequential variant (SNPSE). Our method, inspired by the remarkable success of score-based generative models (Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020), utilises a conditional score-based diffusion model to generate samples from the posterior of interest. While similar approaches (e.g., Batzolis et al., 2021; Dhariwal & Nichol, 2021; Song et al., 2021; Tashiro et al., 2021; Chao et al., 2022; Chung & Ye, 2022) have previously found success in a variety of problems, their application to SBI has not yet been widely investigated.[1]

In contrast to existing SBI approaches based on normalising flows (e.g., SNLE, SNPE), our approach only requires estimates for the gradient of the log density, or score function, of the intractable likelihood or the posterior, which can be approximated using a neural network via score matching techniques (Hyvärinen, 2005; Vincent, 2011; Song et al., 2020). Since we do not require a normalisable model, our method avoids the need for any strong restrictions on the model architecture. In addition, unlike methods based on generative adversarial networks (e.g., GATSBI), we do not require adversarial training objectives, which are notoriously unstable (Metz et al., 2017; Salimans et al., 2016).

We first discuss how conditional score-based diffusion models can be used for SBI. We then outline how our approach can be embedded within a principled sequential training procedure, which guides simulations towards informative regions using the current approximation of the posterior. We outline in detail a number of possible sequential procedures, several of which could also be used to develop sequential variants of amortised algorithms more recently proposed in the SBI literature (e.g., Dax et al., 2023). We then advocate for our preferred method, Truncated Sequential NPSE (TSNPSE), which uses a series of truncated proposals inspired by the approach in Deistler et al. (2022a). We validate our methods on several benchmark SBI problems as well as a real-world neuroscience problem, obtaining comparable or superior performance to other state-of-the-art methods.

2. Simulation-Based Inference with Diffusion Models

2.1. Simulation-Based Inference

Suppose that we have access to a simulator which, given input parameters $\theta \in \mathbb{R}^d$, generates synthetic data $x \in \mathbb{R}^p$. We assume that 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 generating samples from the posterior distribution $p(\theta|x_{\text{obs}}) \propto p(\theta)p(x_{\text{obs}}|\theta)$, given a finite number of i.i.d. samples $\{(\theta_i, x_i)\}_{i=1}^N \sim p(\theta)p(x|\theta)$.

2.2. Diffusion Models for Simulation-Based Inference

We propose to tackle this problem using conditional score-based diffusion models (e.g., Song et al., 2021). In such models, noise is gradually added to the target distribution using a diffusion process, resulting in a tractable reference distribution, e.g., a standard Gaussian. The time-reversal of this process is also a diffusion process, whose dynamics can be approximated using score matching (Hyvärinen, 2005; Vincent, 2011; Song & Ermon, 2020; Song et al., 2021). One can thus generate samples from the target distribution by simulating the approximate reverse-time process, initialised at samples from the reference distribution.

More concretely, we begin by defining a forward noising

[1] In parallel with an early version of this work, Geffner et al. (2023) also studied the use of diffusion models for SBI. We provide a comparison with this paper in Section 4.3 and Appendix D.

process $(\theta_t)_{t\in[0,T]}$ which, initialised at $\theta_0 \sim p(\cdot|x)$, evolves according to the stochastic differential equation (SDE)

$$\mathrm{d}\theta_t = f(\theta_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t, \tag{2}$$

where $f : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ is the drift coefficient, $g : \mathbb{R}_+ \to \mathbb{R}^d$ is the diffusion coefficient, and $(w_t)_{t\geq 0}$ is a standard $\mathbb{R}^d$-valued Brownian motion. The coefficients f and g are chosen such that, for all $x \in \mathbb{R}^p$, the forward noising process admits a unique stationary distribution π from which it is easy to sample, e.g., a standard Gaussian.

Under mild conditions, the time-reversed process $(\bar{\theta}_t)_{t\in[0,T]} := (\theta_{T-t})_{t\in[0,T]}$ is also a diffusion process (Anderson, 1982; Föllmer, 1985; Haussmann & Pardoux, 1986). Initialised at $\bar{\theta}_0 \sim p_T(\cdot|x)$, this process evolves according to

$$\mathrm{d}\bar{\theta}_t = \left[-f(\bar{\theta}_t, T-t) + g^2(T-t)\nabla_\theta \log p_{T-t}(\bar{\theta}_t|x)\right]\mathrm{d}t + g(T-t)\mathrm{d}w_t, \tag{3}$$

where $p_t(\cdot|x) = \int p_{t|0}(\cdot|\theta_0)p(\theta_0|x)\mathrm{d}\theta_0$ denotes the time marginal density of θ_t, conditioned on x. By definition, the marginals of $(\bar{\theta}_t)_{t\in[0,T]}|x$ are equal to those of $(\theta_{T-t})_{t\in[0,T]}|x$. Thus, in particular, $\bar{\theta}_T \sim p_0(\cdot|x) := p(\cdot|x)$. Hence, if we could sample $\bar{\theta}_0 \sim p_T(\cdot|x)$, and simulate $(\bar{\theta}_t)_{t\in[0,T]}$ according to (3), then its final distribution would be the desired posterior distribution. This process is visualised in Figure 1.

Although this procedure provides an elegant sampling mechanism, it does not allow us to evaluate the density $p_0(\theta|x) := p(\theta|x)$ of these samples. Fortunately, there exists an ODE with the same marginals as (2), which does enable density evaluation. This deterministic process, known as the probability flow ODE (Song et al., 2021), defines $(\theta_t)_{t\in[0,T]}$ according to

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = \left[f(\theta_t, t) - \frac{1}{2}g^2(t)\nabla_\theta \log p_t(\theta_t|x)\right], \tag{4}$$

where once again $\theta_0 \sim p(\cdot|x)$. In this case, the log densities $\log p_t(\theta_t|x)$ *can* be computed exactly via the instantaneous change-of-variables formula (Chen et al., 2018a):

$$\frac{\mathrm{d}\log p_t(\theta_t|x)}{\mathrm{d}t} = -\mathrm{Tr}\left[\nabla_\theta\big(f(\theta_t, t) - \frac{1}{2}g^2(t)\nabla_\theta \log p_t(\theta_t|x)\big)\right]. \tag{5}$$

In practice, we cannot simulate (3) or (4) directly, since we do not have access to $p_T(\cdot|x)$, or the scores $\nabla_\theta \log p_t(\theta_t|x)$. We will therefore rely on two approximations. First, we will assume that $p_T \approx \pi$. Second, we will approximate $\nabla_\theta \log p_t(\theta_t|x)$ using score matching (e.g., Song et al., 2021), and substitute this approximation into (3) or (4). In this case, the ODE in (4) is an instance of a continuous normalising flow (CNF) (Grathwohl et al., 2019).

There are various ways in which we can obtain this approximation. Here, we choose to train a time-varying score network $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ to directly approximate the score of the perturbed posterior (Dhariwal & Nichol, 2021; Song et al., 2021; Batzolis et al., 2021).[2] In this case, a natural objective is the weighted Fisher divergence

$$\mathcal{J}^{\mathrm{SM}}_{\mathrm{post}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \, \mathbb{E}_{p_t(\theta_t,x)}\left[||s_\psi(\theta_t, x, t) - \nabla_\theta \log p_t(\theta_t|x)||^2\right]\mathrm{d}t, \tag{6}$$

where $\lambda_t : [0,T] \to \mathbb{R}_+$ is a positive weighting function, and $p_t(\theta_t, x)$ denotes the joint distribution of (θ_t, x). In practice, this objective cannot be evaluated directly, since it depends on the posterior scores $\nabla_\theta \log p_t(\theta_t|x)$. Fortunately, one can show (e.g., Batzolis et al., 2021; Tashiro et al., 2021; Appendix A.1) that it is equivalent to minimise the conditional denoising posterior score matching objective, given by

$$\mathcal{J}^{\mathrm{DSM}}_{\mathrm{post}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)p(\theta_0)} \left[||s_\psi(\theta_t, x, t) - \nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)||^2\right]\mathrm{d}t, \tag{7}$$

where $p_{t|0}(\theta_t|\theta_0)$ denotes the transition density defined by (2). In particular, this objective is minimised when $s_\psi(\theta_t, x, t) = \nabla_\theta \log p_t(\theta_t|x)$ for almost all $\theta_t \in \mathbb{R}^d$, $x \in \mathbb{R}^p$, and $t \in [0,T]$.

The expectation in (7) only depends on samples $\theta_0 \sim p(\theta)$ from the prior, $x \sim p(x|\theta_0)$ from the simulator, and $\theta_t \sim p_{t|0}(\theta_t|\theta_0)$ from the forward diffusion (2). Moreover, given a suitable choice for the drift and diffusion coefficients in (2), the scores $\nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)$ can be computed in closed form. We can thus compute a Monte Carlo estimate of (7), and minimise this to obtain $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$.

We now have all of the necessary ingredients to generate approximate samples from the target posterior distribution:

(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 process (2).

(ii) Using these samples, train a time-varying score network $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ by minimising a Monte Carlo estimate of (7).

(iii) Draw samples $\bar{\theta}_0 \sim \pi(\cdot)$. Simulate an approximation of the reverse-time process in (3), or the time-reversal of the probability flow ODE in (4), with $x = x_{\mathrm{obs}}$, replacing $\nabla_\theta \log p_t(\theta_t|x_{\mathrm{obs}}) \approx s_\psi(\theta_t, x_{\mathrm{obs}}, t)$.

[2] In Appendix B, we outline an alternative approach which instead trains a score-network to approximate the score of the perturbed likelihood $\nabla_\theta \log p_t(x|\theta_t)$. We refer to this approach as Neural Likelihood Score Estimation (NLSE).

In line with the current SBI taxonomy, we will refer to this approach as Neural Posterior Score Estimation (NPSE).

In Appendix A.2, we provide error bounds for NPSE in the fully deterministic sampling regime, assuming an L^2 bound on the approximation error and a mild regularity condition on the target posterior $p(\cdot|x_{\text{obs}})$. Our result is adapted from Benton et al. (2024, Theorem 6).

3. Sequential Neural Score Estimation

Given enough data and a sufficiently flexible model, the optimal score network $s_{\psi^*}(\theta_t, x, t)$ will equal $\nabla_\theta \log p_t(\theta_t|x)$ for almost all $x \in \mathbb{R}^p$, $\theta_t \in \mathbb{R}^d$, and $t \in [0, T]$. Thus, in theory, we can use the methods in the previous section to generate samples $\theta \sim p(\theta|x)$ for any observation x.

In practice, we are often only interested in sampling from the posterior for a particular experimental observation $x = x_{\text{obs}}$. Thus, given a finite simulation budget, it may be more efficient to train the score network using simulated data which is close to x_{obs}, and thus more informative for learning the posterior scores $\nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$. This can be achieved by drawing initial parameter samples from a suitably chosen proposal prior, $\theta_0 \sim \tilde{p}(\theta)$, rather than the true prior $\theta_0 \sim p(\theta)$. This idea is central to existing sequential SBI algorithms, which use a sequence of adaptively chosen proposals in order to guide simulations towards more informative regions. The central challenge associated with developing a successful sequential algorithm is how to effectively correct for the mismatch between the so-called proposal posterior

$$\tilde{p}(\theta|x) = p(\theta|x)\frac{\tilde{p}(\theta)}{p(\theta)}\frac{p(x)}{\tilde{p}(x)}, \tag{8}$$

and the true posterior $p(\theta|x) \propto p(\theta)p(x|\theta)$. In the following sections, we introduce several possible sequential variants of NPSE, which we collectively refer to as SNPSE. We note, as pointed out in the introduction, that in principle these approaches could also be used to develop sequential variants of the recently proposed flow-matching posterior estimation (FMPE) algorithm (Dax et al., 2023).

We begin by outlining some generic features of the sequential procedure, which hold irrespective of the specific sequential method employed (see Sections 3.1 - 3.2). In all cases, the sequential procedure will take place over R rounds, indexed by $r \geq 1$. Given a total budget of N simulations, we assume the simulations are evenly distributed across rounds: $N_r = N/R = M$ for $r = 1, \ldots, R$, where N_r is the number of simulations in round r. In the first round, we follow the standard NPSE algorithm (Section 2). In particular, we first generate $\{\theta_{0,i}^1\}_{i=1}^M \sim p(\theta)$ from the prior, and $\{x_i^1\}_{i=1}^M \sim p(x|\theta_{0,i})$ using the simulator. These samples are used to train a score network $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ by minimising (7). By substituting this into (3), we can generate samples approximately from the target posterior.

Following the initial round, there are several conceivable sequential procedures one could use to generate samples from $p(\theta|x_{\text{obs}})$. We now describe several such methods. Broadly speaking, these procedures differ in (i) how they define the proposal prior; and (ii) how they correct for the mismatch between the proposal posterior and the true posterior.

3.1. Truncated Approach

We first introduce our preferred method: Truncated SNPSE (TSNPSE). This algorithm - summarised in Algorithm 1 - utilises a series of proposals given by truncated versions of the prior, inspired by the approach in Deistler et al. (2022a). For $r \geq 1$, let $p_\psi^{r-1}(\theta|x_{\text{obs}})$ denote the approximation to the target posterior learned in the $(r-1)^{\text{th}}$ round, with the convention that $p_\psi^0(\theta) := p(\theta)$. Then, in the r^{th} round, we will use the highest-probability region of this approximation to define a truncated version of the prior. To be precise, in the r^{th} round, suppose we define

$$\bar{p}^r(\theta) \propto p(\theta) \cdot \mathbb{I}\{\theta \in \text{HPR}_\varepsilon(p_\psi^{r-1}(\theta|x_{\text{obs}}))\}, \tag{9}$$

where $\text{HPR}_\varepsilon(\cdot)$ denotes the highest $1-\varepsilon$ probability region, defined as the smallest region which contains $1-\varepsilon$ of the mass; and we adopt the convention that $\bar{p}^0(\theta) = p(\theta)$. We then define the proposal distribution for this round as $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1}\bar{p}^s(\theta)$. Additional details regarding how to compute and sample from this proposal distribution are provided in Appendix E.3.

Crucially, under the assumption that we do not truncate regions which have non-zero mass under the true posterior $p(\theta|x_{\text{obs}})$, this proposal distribution is proportional to the prior within the support of the posterior. Thus, we do not need to perform a correction. In particular, our loss function remains minimised at the score of the target posterior. This statement is formalised in the following proposition.

Proposition 3.1. *Let $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1}\bar{p}^s(\theta)$, where $\bar{p}^0(\theta) = p(\theta)$ and $\bar{p}^s(\theta)$ is defined by (9) for all $s \geq 1$. Suppose that $\Theta_{\text{obs}} \subseteq \text{HPR}_\epsilon(p_\psi^s(\theta|x_{\text{obs}}))$ for all $s \geq 1$, where $\Theta_{\text{obs}} = \text{supp}(p(\cdot|x_{\text{obs}}))$. Then, writing $\tilde{p}_t^r(\theta_t, x)$ for the distribution of (θ_t, x) when $(\theta_0, x) \sim \tilde{p}^r(\theta, x)$, the minimiser ψ^* of the loss function*

$$\mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{SM}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{\tilde{p}_t^r(\theta_t, x)} \left[||s_\psi(\theta_t, x, t) - \nabla_\theta \log p_t(\theta_t|x)||^2\right]\text{d}t, \tag{10}$$

or, equivalently, of the loss function

$$\mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{DSM}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)} \left[||s_\psi(\theta_t, x, t) - \nabla_\theta \log p_{t|0}(\theta_t|\theta_0)||^2\right]\text{d}t, \tag{11}$$

satisfies $s_{\psi^\star}(\theta_t, x_{\text{obs}}, t) = \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$.

Proof. See Appendix C.1. □

Algorithm 1 TSNPSE

```
Inputs: Observation $x_{\text{obs}}$, prior $p(\theta) =: \bar{p}^0(\theta)$, simulator $p(x|\theta)$, simulation budget $N$, number of rounds $R$, (simulations-per-round $M = N/R$), dataset $\mathcal{D} = \{\}$.
Outputs: $p_\psi(\theta|x_{\text{obs}}) \approx p(\theta|x_{\text{obs}})$.
for $r = 1, \ldots, R$ do
    for $i = 1, \ldots, M$ do
        Draw $\theta_i \sim \bar{p}^{r-1}(\theta)$, $x_i \sim p(x|\theta_i)$.
        Add $(\theta_i, x_i)$ to $\mathcal{D}$.
    end for
    Learn $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ by minimising a Monte Carlo estimate of (11) based on dataset $\mathcal{D}$.
    Compute $\bar{p}^r(\theta)$ in (9) using $s_\psi(\theta_t, x_{\text{obs}}, t)$. See Appendix E.3 for details.
end for
Get $p_\psi(\theta|x_{\text{obs}})$ sampler by substituting $s_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ in (4).
Return: $p_\psi(\theta|x_{\text{obs}})$.
```

3.2. Alternative Approaches

We now outline several other possible sequential approaches for NPSE. An extensive and detailed discussion of these methods, as well as supporting numerical results, can be found in Appendix C. Broadly speaking, these methods can be viewed as score-based analogues of existing sequential variants of NPE, namely, SNPE-A (Papamakarios & Murray, 2016), SNPE-B (Lueckmann et al., 2017), and SNPE-C (Greenberg et al., 2019). We refer to, e.g., Durkan et al. (2020) for a concise overview of SNPE-A, SNPE-B, and SNPE-C.

Unlike TSNPSE, in each of these methods, the proposal prior is defined *directly* in terms of the most recent approximation of the posterior. In particular, in the r^{th} round, we now sample new parameters $\{\theta_{0,i}^r\}_{i=1}^M \sim p_\psi^{r-1}(\theta|x_{\text{obs}})$ and simulate new data $\{x_i^r\}_{i=1}^M \sim p(x|\theta_{0,i}^r)$. We then concatenate these samples with those from previous rounds to form $\bigcup_{s=1}^r \{(\theta_{0,i}^s, x_i^s)\}_{i=1}^M \sim \tilde{p}^r(\theta)p(x|\theta)$, where $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1} p_\psi^s(\theta|x_{\text{obs}})$, and $p_\psi^0(\theta|x_{\text{obs}}) := p(\theta)$.

In this case, if were to minimise the original score matching objective (7), but using samples $\theta_0 \sim \tilde{p}^r(\theta)$ rather than $\theta_0 \sim p(\theta)$, we would learn a score network which approximates $\nabla_\theta \log \tilde{p}_t^r(\theta_t|x)$, rather than $\nabla_\theta \log p_t(\theta_t|x)$, where $\tilde{p}_t^r(\theta_t|x) = \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0|x)\mathrm{d}\theta_0$, and $\tilde{p}^r(\theta|x) = \frac{\tilde{p}^r(\theta)p(x|\theta)}{\tilde{p}^r(x)}$. Substituting this score network, evaluated at $x = x_{\text{obs}}$, into (3) or (4), would then result in samples $\theta \sim \tilde{p}^r(\theta|x_{\text{obs}})$, rather than $\theta \sim p(\theta|x_{\text{obs}})$. We thus require a correction to recover samples from the correct posterior.

SNPSE-A. The first approach is to perform a post-hoc importance weight correction using, e.g., sampling-importance resampling (SIR) (Rubin, 1987; 1988; Smith & Gelfand, 1992; Gelman et al., 1995). According to this approach, we first generate $\{\tilde{\theta}_i\}_{i=1}^{M'} \sim \tilde{p}_\psi^r(\cdot|x_{\text{obs}})$, where $\tilde{p}_\psi^r(\cdot|x_{\text{obs}})$ denotes the approximate proposal posterior obtained in the r^{th} round, and $M' \geq M$. We then draw samples $\{\theta_i\}_{i=1}^M$ with or without replacement from $\{\tilde{\theta}_i\}_{i=1}^{M'}$, with sample probabilities, $\tilde{w}_i$, proportional to the importance ratios

$$\tilde{h}_i = \frac{p(\tilde{\theta}_i|x_{\text{obs}})}{\tilde{p}_\psi^r(\tilde{\theta}_i|x_{\text{obs}})}. \quad (12)$$

In the limit as $M' \to \infty$, this sample will consist of independent draws from $p(\cdot|x_{\text{obs}})$ (e.g., Smith & Gelfand, 1992). In practice, we cannot evaluate $p(\cdot|x_{\text{obs}})$ in (12), and thus will instead use sample probabilities w_i proportional to

$$h_i = \frac{p(\tilde{\theta}_i)}{\tilde{p}^r(\tilde{\theta}_i)}. \quad (13)$$

The importance ratios in (13) are approximately proportional to the correct importance ratios in (12), since

$$h_i = \frac{p(\tilde{\theta}_i)}{\tilde{p}^r(\tilde{\theta}_i)} \propto \frac{p(\tilde{\theta}_i|x_{\text{obs}})}{\tilde{p}^r(\tilde{\theta}_i|x_{\text{obs}})} \approx \frac{p(\tilde{\theta}_i|x_{\text{obs}})}{\tilde{p}_\psi^r(\tilde{\theta}_i|x_{\text{obs}})} = \tilde{h}_i. \quad (14)$$

Although SNPSE-A can work well in simple settings, it is fundamentally limited by the approximation introduced in (14). In particular, when there is a significant mismatch between the true proposal, $\tilde{p}^r(\cdot|x_{\text{obs}})$, and the approximate (learned) proposal, $\tilde{p}_\psi^r(\cdot|x_{\text{obs}})$, this approach can lead to inaccurate inference (see Appendix C.2).

SNPSE-B. The second approach is to include an importance weight correction within the denoising score matching objective (7). In particular, in the r^{th} round, we now minimise a Monte Carlo estimate of

$$\mathcal{J}_{\text{post}}^{\text{SNPSE-B}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)} \quad (15)$$

$$\left[\frac{p(\theta_0)}{\tilde{p}^r(\theta_0)}||s_\psi(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)||^2\right]\mathrm{d}t.$$

It is straightforward to show that this objective is minimised at the score of the true posterior, that is, by ψ^* such that $s_{\psi^*}(\theta_t, x, t) = \nabla_\theta \log p_t(\theta_t|x)$ (see Appendix C.3). Unfortunately, similar to SNPE-B (Lueckmann et al., 2017), the importance weights are often high variance, resulting in unstable training and poor overall algorithm performance (e.g., Papamakarios et al., 2019; Durkan et al., 2019).

SNPSE-C. The third approach is to include a score-based correction within the denoising posterior score matching objective (7). In this case, we minimise (7), now over samples from the proposal prior, to learn an estimate $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$ of the proposal posterior. We would like to use this to automatically recover an estimate of $\nabla_\theta \log p_t(\theta_t|x)$. To do so, observe that

$$\nabla_\theta \log p_t(\theta_t|x) = \nabla_\theta \log p_t(\theta_t) + \nabla_\theta \log p_t(x|\theta_t) \tag{16}$$

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t|x) = \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) \tag{17}$$

where $p_t(x|\theta_t) = \int p(x|\theta_0)p_{0|t}(\theta_0|\theta_t)\mathrm{d}\theta_0$ and $\tilde{p}^r_t(x|\theta_t) = \int p(x|\theta_0)\tilde{p}^r_{0|t}(\theta_0|\theta_t)\mathrm{d}\theta_0$. Thus, in particular,

$$\begin{aligned}\nabla_\theta \log \tilde{p}^r_t(\theta_t|x) = \nabla_\theta &\log p_t(\theta_t|x) \\ &+ \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) \\ &- \nabla_\theta \log p_t(\theta_t) - \nabla_\theta \log p_t(x|\theta_t).\end{aligned} \tag{18}$$

This identity suggests defining $\tilde{s}^r_\psi(\theta_t, x, t)$ in terms of another score network $s_\psi(\theta_t, x, t)$ according to

$$\begin{aligned}\tilde{s}^r_\psi(\theta_t, x, t) = s_\psi&(\theta_t, x, t) \\ &+ \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) \\ &- \nabla_\theta \log p_t(\theta_t) - \nabla_\theta \log p_t(x|\theta_t).\end{aligned} \tag{19}$$

In this case, given $\tilde{s}^r_\psi(\theta_t, x_{\mathrm{obs}}, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x_{\mathrm{obs}})$, we also have $s_\psi(\theta_t, x_{\mathrm{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\mathrm{obs}})$ from (18) - (19), as required. Unlike SNPSE-A and SNPSE-B, SNPSE-C has the advantage of not requiring importance weights. Moreover, since the corrections are performed 'in the score space', it does not require us to evaluate $\tilde{p}^r(\cdot)$, and thus does not necessitate calculating likelihoods via (4) and (5). On the other hand, it does require knowledge of $\nabla_\theta \log \tilde{p}^r_t(\theta_t)$, $\nabla_\theta \log \tilde{p}^r_t(x|\theta_t)$, $\nabla_\theta \log p_t(\theta_t)$, and $\nabla_\theta \log p_t(x|\theta_t)$, which are not immediately available. Thus, in practice, this approach depends on several additional approximations. We provide further details in Appendix C.4.

In empirical testing, the corrections required for SNPSE-A, SNPSE-B, and SNPSE-C, lead to significantly worse performance than TSNPSE (see Appendix C). On this basis, we advocate for TSNPSE as the preferred sequential method, and focus exclusively on this approach in our subsequent numerics (see Section 5).

4. Related Work

4.1. Simulation-Based Inference

Approximating the Posterior. Many modern SBI algorithms are based on learning a conditional neural density estimator $q_\psi(\theta|x)$ to approximate the posterior $p(\theta|x)$, often over a number of rounds of training (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019). This approach is known as SNPE. Such methods circumvent the bias introduced by the use of a proposal prior in various ways, including a post-hoc importance weight correction (SNPE-A) (Papamakarios & Murray, 2016), minimising an importance weighted loss function (SNPE-B) (Lueckmann et al., 2017), and re-parametrising the proposal posterior objective (SNPE-C) (Greenberg et al., 2019). Alternatively, the use of a truncated prior as the proposal circumvents the need for a correction (TNPSE) (Deistler et al., 2022a). As noted in Section 3, our sequential methods can loosely be viewed as analogues of these approaches suitable for diffusion models.

Approximating the Likelihood. Rather than approximating the posterior directly, another approach is to learn a model $q_\psi(x|\theta)$ for the intractable likelihood $p(x|\theta)$. Such methods are sometimes referred to as Synthetic Likelihood approaches (Wood, 2010; Ong et al., 2018; Price et al., 2018; Frazier et al., 2022). Early examples of this approach assume that the likelihood can be parameterised as a single Gaussian (Wood, 2010), or a mixture of Gaussians (Fan et al., 2013). More recent approaches, referred to as SNLE, train conditional neural density estimators, over a number of rounds (Lueckmann et al., 2019; Papamakarios et al., 2019). While SNLE does not require a correction, it does rely on MCMC to generate posterior samples. This can be costly, and may prove prohibitive for posteriors with complex geometries.

Approximating the Likelihood Ratio. Another approach to simulation-based inference is based on learning a parametric model for the likelihood-to-marginal ratio $r(x, \theta) = p(x|\theta)/p(x) = p(\theta|x)/p(\theta)$ (Izbicki et al., 2014; Tran et al., 2017; Durkan et al., 2020; Hermans et al., 2020; Miller et al., 2021; Simons et al., 2021; Thomas et al., 2022), or the likelihood ratio $r(x, \theta_1, \theta_2) = p(x|\theta_1)/p(x|\theta_2)$ (Pham et al., 2014; Cranmer et al., 2016; Gutmann et al., 2018; Stoye et al., 2019; Brehmer et al., 2020). In the first case, one trains a binary classifier to approximate this ratio. Using the fact that $p(\theta|x_{\mathrm{obs}}) = p(\theta)r(x_{\mathrm{obs}}, \theta)$, one can then use MCMC to generate posterior samples. This approach is also amenable to a sequential implementation, known as SNRE (Durkan et al., 2020).

Approximating the Posterior and the Likelihood. Two recent methods aim to combine the advantages of SNLE (or SNRE) and SNPE, while addressing their shortcomings (Wiqvist et al., 2021; Glockler et al., 2022). In particular, SNVI (Glockler et al., 2022) and Sequential Neural Posterior and Likelihood Approximation (SNPLA) (Wiqvist et al., 2021) first train a neural density estimator $q_{\psi_{\mathrm{lik}}}(x|\theta)$ to approximate the likelihood, or the likelihood ratio. Once this model has been trained, one trains a parametric approximation $q_{\psi_{\mathrm{post}}}(\theta)$ for the posterior, using variational inference with normalising flows. These methods differ in their variational objectives: SNVI uses the forward KL divergence, the importance weighted ELBO, or the Renyi α-divergence, while SNPLA uses the reverse KL divergence.

4.2. Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) have recently emerged as a new class of generative models. These models offer high quality generation and sample diversity, do not require adversarial training, and have achieved state-of-the-art performance in a range of applications, including image generation (Dhariwal & Nichol, 2021; Ho et al., 2020; Song et al., 2021), audio synthesis (Chen et al., 2021; Kong et al., 2021; Popov et al., 2021), shape generation (Cai et al., 2020), music generation (Mittal et al., 2021), and video generation (Ho et al., 2022).

Conditional diffusion models (Song & Ermon, 2019; Song et al., 2021; Batzolis et al., 2021; Dhariwal & Nichol, 2021; Chao et al., 2022) extend this framework to allow for conditional generation, allowing for tasks such as image inpainting (Song et al., 2021), time series imputation (Tashiro et al., 2021), image colourisation (Song et al., 2021), and medical image reconstruction (Song et al., 2022). In such applications, the 'prior' typically corresponds to an unknown data distribution, whose score is estimated using score matching. Meanwhile, the 'likelihood' is often known, or else corresponds to a differentiable classifier (e.g., Song et al., 2021). This is rather different to our setting, in which the prior is typically known, while the likelihood is intractable.

4.3. Diffusion Models and Simulation-Based Inference

Surprisingly, the application of diffusion models to problems of interest to the SBI community (see, e.g., Lueckmann et al., 2021) has not previously been investigated. In parallel with this work, Geffner et al. (2023) also considered the use of diffusion models for SBI. While related to our work, Geffner et al. (2023) focused specifically on how to use NPSE for sampling from $p(\theta|x^1_{\text{obs}},\dots,x^n_{\text{obs}})$, for any set of observations $\{x^1_{\text{obs}},\dots,x^n_{\text{obs}}\}$. Meanwhile, we introduce sequential variant(s) of NPSE (see Section 3). We provide a more detailed comparison with this work in Appendix D.

More recently, several other authors have proposed SBI algorithms which are closely related to diffusion models. In particular, Dax et al. (2023) propose flow matching posterior estimation (FMPE), an SBI algorithm which approximates the posterior $p(\theta|x)$ using a CNF trained via flow matching (Lipman et al., 2023). This approach includes NPSE, when using the deterministic probability-flow ODE, as a special case. Meanwhile, Schmitt et al. (2023) introduce consistency model posterior estimation (CMPE), which applies consistency models (Song et al., 2023) to SBI. In contrast to our work, both of these papers consider only the amortised setting, and do not introduce sequential variants of their algorithms.

5. Numerical Experiments

In this section we benchmark the numerical performance of NPSE and TSNPSE. Code to reproduce our numerical results can be found at `https://github.com/jacksimons15327/snpse_icml`.

5.1. Experimental Details

In all experiments, our score network is comprised of independent multilayer perceptron (MLP) embedding networks for θ_t and x. A sinusoidal embedding is employed for t. The embeddings of θ_t, x, t are concatenated and input to a MLP. All MLP networks have 3 fully connected layers, each with 256 neurons and SiLU activation functions. We use Adam (Kingma & Ba, 2015) to train the networks, with a learning rate of 10^{-4}. We hold back 15% of the data to be used as a validation set for early stopping. We provide details of any additional hyperparameters in Appendix E.3.2.

5.2. Benchmark Results

We first provide results for eight popular SBI benchmarks described in Lueckmann et al. (2021) (see Appendix E.1 for details). We consider simulation budgets of 1000, 10000 and 100000. In all cases, we report the classification-based two-sample test (C2ST) score (Lopez-Paz & Oquab, 2017), which varies between 0.5 and 1 (lower is better), with a score of 0.5 indicating perfect posterior estimation.

For both our non-sequential (NPSE) and sequential (TSNPSE) methods, we consider two choices of dynamics for the forward noising process: a variance-exploding SDE (VE SDE) and a variance-preserving SDE (VP SDE) (Song et al., 2021). Further details can be found in Appendix E.3.1. For reference, we compare our non-sequential method (NPSE) with NPE (Papamakarios & Murray, 2016); and our sequential method (TSNPSE) with SNPE-C (Greenberg et al., 2019) and TSNPE (Deistler et al., 2022a). For these algorithms, we obtain results using the Python toolkit `sbibm` (Lueckmann et al., 2021). We include an additional comparison with FMPE (Dax et al., 2023) in Appendix F.

Our results, provided in Figures 2 and 3, demonstrate that diffusion models provide an accurate and robust alternative to state-of-the-art SBI methods based on posterior density estimation with (discrete) normalising flows. Notably, for the two most challenging benchmark experiments, SLCP and Lotka Volterra, our methods outperform their competitors, providing evidence that our proposed algorithms scale well to high-dimensions. For the remaining benchmark experiments, the results are more mixed, with the best performing method varying based on the task at hand as well as the simulation budget. It is worth emphasising that our algorithms employ the same hyperparameter settings (e.g., neural network architecture, optimizer, etc.) across all exper-

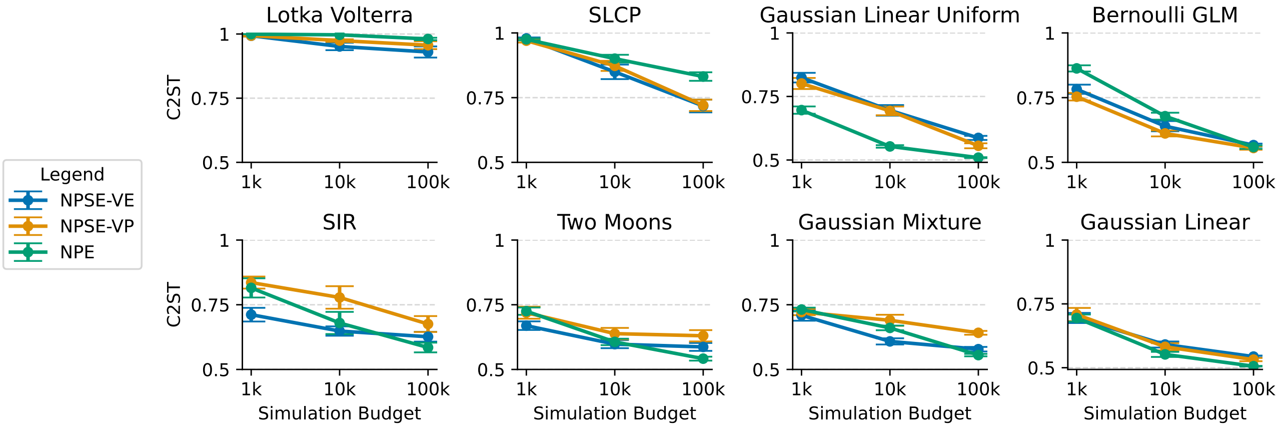


Figure 2. **Results on eight benchmark tasks (non-sequential methods).**

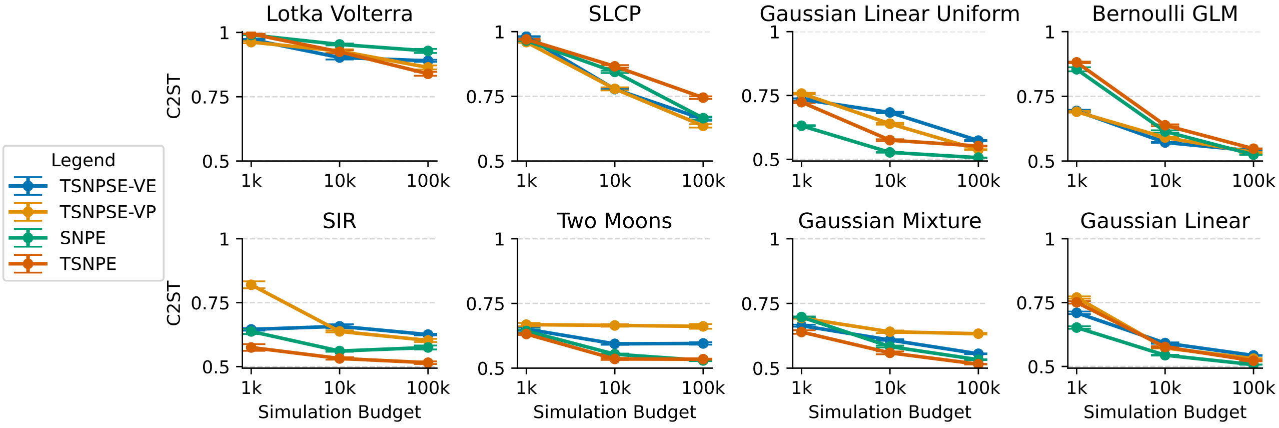


Figure 3. **Results on eight benchmark tasks (sequential methods).**

iments, including both the benchmarks and the real-world experiment in Section 5.3, and that we did not perform an extensive hyperparameter search. We suspect that the performance of (TS)NPSE could be further improved with additional tuning.

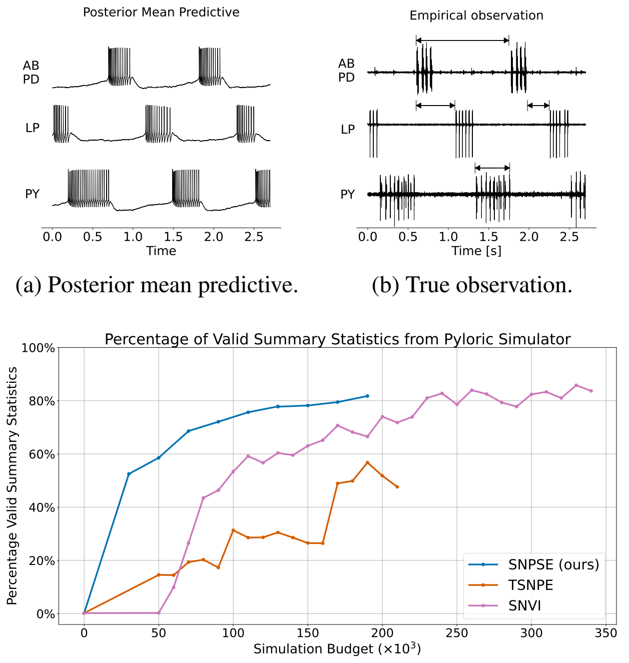
We also note that the choice of dynamics (e.g., VE SDE or VP SDE) for the forward noising process can have a significant impact on the quality of the posterior inference, although the best performing method can fluctuate based on the task at hand. In general, based on our empirical results, we recommend VE SDE for low dimensional experiments, and VP SDE for high dimensional experiments.

5.3. Real-world Neuroscience Problem

We also apply TSNPSE to a challenging real-world neuroscience problem: inference for the parameters of a simulator model of the pyloric network of the stomatogastric ganglion in the crab *Cancer borealis* (Prinz et al., 2003; 2004). In this case, the model simulates 3 neurons, whose behaviours are governed by synapses and membrane conductances which together constitute a set of 31 parameters. The simulator outputs 3 voltage traces, which are condensed into 18 summary statistics (Prinz et al., 2003; 2004). The prior is uniform over previously defined parameter ranges (Prinz et al., 2004; Gonçalves et al., 2020). We are interested in inferring the posterior distribution of the parameters, given experimentally observed data (Haddad & Marder, 2021).

In this model, the volume of the parameter space which gives rise to meaningful summary statistics is very small. For example, over 99% of prior samples input into the simulator result in neural traces with ill-defined summary statistics. This, alongside the significant simulator cost, renders posterior inference in this model a very challenging task. Previous work has performed amortised inference using NPE, although this requires several million simulations (Gonçalves et al., 2020; Deistler et al., 2022b). More recent methods have adopted a sequential approach, reducing the number of samples required by 25 times or more (Glockler et al., 2022; Deistler et al., 2022a; Glaser et al., 2022).

We applied TSNPSE to this problem, using an identical architecture to that used in our benchmark experiments to demonstrate the robustness of our approach. We performed



(a) Posterior mean predictive.

(b) True observation.

(c) Percentage of valid summary statistics from the pyloric simulator against simulation budget for 3 different methods, SNPSE (ours, blue), TSNPE (orange), SNVI (pink).

Figure 4. **Results for the Pyloric experiment.**

inference over 9 rounds, with 30000 initial simulations, and 20000 added simulations in each round. Our results, including the percentage of valid summary statistics versus the number of simulations, and a posterior predictive sample, are provided in Figure 4. We also provide a pairwise marginal plot of our final posterior approximation in Figure 7 (Appendix E.2). In the final round, we achieved 81% valid summary statistics from the simulator (Figure 4c), superior to the percentage achieved by other methods for the same simulation budget. We also note that the obtained posterior produces samples which closely match the observed data (Figure 4a). In addition, the posterior marginals (Figure 7) are very similar to others previously reported in the literature (Deistler et al., 2022a; Glockler et al., 2022).

6. Discussion

Limitations The main limitation of our approach relates to computational cost. In particular, TSNPSE requires computing $\mathrm{HPR}_{\varepsilon}$ of the approximate posterior to define the proposal, which involves computing the approximate posterior density over many samples. In TSNPE, which uses a normalising flow, this is relatively inexpensive as a single forward pass is required for sampling, and a single backward pass for density evaluation (Dinh et al., 2017; Papamakarios et al., 2017). In contrast, with TSNPSE, which uses a CNF, multiple forward passes are required for sampling, and multiple gradients of forward passes are required for density evaluation (Chen et al., 2018b; Grathwohl et al., 2019). Interestingly, this can be avoided using an alternative parameterisation of the score network; see Appendix G for further details.

It is worth noting that there are several ways to reduce the cost of both sampling and likelihood evaluation in CNFs. For example, faster numerical ODE solvers can substantially reduce the number of forward passes required for sampling (e.g., Lu et al., 2022; Zhang & Chen, 2023). Meanwhile, the Skilling-Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1990) can be used to reduce the cost of the gradient computations required for likelihood evaluation (Grathwohl et al., 2019).

In some sense, this comparison between TSNPE and TSNPSE reflects a wider discussion regarding the trade-offs between (discrete) normalising flows and CNFs. The former are associated with a lower computational cost, while the latter are much more flexible, which can result in more accurate inference (e.g., Grathwohl et al., 2019; Finlay et al., 2020). As such, preference for a method based on a discrete normalising flow (e.g., TSNPE) or a CNF (e.g., TSNPSE) will depend on the problem at hand. For example, for challenging real-world simulators, the additional cost incurred by a CNF may be negligible in comparison to the cost of acquiring simulations.

Future Work We highlight two directions for future work. First, with the exception of SNPSE-C, the sequential methods in this paper can also be applied to other methods based on CNFs. In this sense, a natural extension of our work would be to develop a sequential variant of FMPE (Dax et al., 2023). Second, in this paper we used a relatively simple neural network architecture, with a relatively small number of parameters, in large part to demonstrate the robustness of our approach. In contrast, the architectures used by diffusion models in other modalities are often highly specialised, and have received significant attention in their own right (e.g., Karras et al., 2022). Undoubtedly, further investigation into effective network design for SBI problems would be a fruitful direction for future work.

Acknowledgements

The authors are grateful to the three anonymous reviewers for their constructive feedback. The authors would also like to thank Tomas Geffner and Iain Murray for useful comments on early drafts of the paper. LS was supported by the UK Research and Innovation (UKRI) Engineering and Physical Sciences Research Council (EPSRC), grant number EP/V022636/1. JS was supported by the EPSRC Centre for Doctoral Training in Computational Statistics and Data Science, grant number EP/S023569/1.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are several potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alsing, J., Wandelt, B., and Feeney, S. 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, 2018. doi: 10.1093/mnras/sty819. 1

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

Bartholomew-Biggs, M., Brown, S., Christianson, B., and Dixon, L. Automatic differentiation of algorithms. *Journal of Computational and Applied Mathematics*, 124(1): 171–190, 2000. doi: 10.1016/S0377-0427(00)00422-2. 21

Batzolis, G., Stanczuk, J., Schönlieb, C.-B., and Etmann, C. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021. 2, 3, 7, 17, 23, 24, 28, 30

Beaumont, M. A. Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41(1):379–406, 2010. doi: 10.1146/annurev-ecolsys-102209-144621. 1

Beaumont, M. A., Zhang, W., and Balding, D. J. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. doi: 10.1093/genetics/162.4.2025. 1

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009. doi: 10.1093/biomet/asp052. 1, 34

Benton, J., Deligiannidis, G., and Doucet, A. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024. 4, 18, 19, 20

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, New York, NY, 2007. 27

Bishop, C. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006. 21

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. 1

Bonassi, F. V. and West, M. Sequential Monte Carlo with adaptive weights for approximate Bayesian computation. *Bayesian Analysis*, 10(1):171–187, 2015. doi: 10.1214/14-BA891. 1

Brehmer, J. Simulation-based inference in particle physics. *Nature Reviews Physics*, 3(5):305, 2021. doi: 10.1038/s42254-021-00305-6. 1

Brehmer, J., Kling, F., Espejo, I., and Cranmer, K. MadMiner: machine learning-based inference for particle physics. *Computing and Software for Big Science*, 4(1): 3, 2020. doi: 10.1007/s41781-020-0035-2. 6

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 1st edition, 2011. ISBN 9781420079418. 1

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., and Hariharan, B. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV 2020)*, Glasgow, UK, 2020. 7

Chao, C.-H., Sun, W.-F., Cheng, B.-W., Lo, Y.-C., Chang, C.-C., Liu, Y.-L., Chang, Y.-L., Chen, C.-P., and Lee, C.-Y. Denoising likelihood score matching for conditional score-based data generation. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022. 2, 7, 21

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. WaveGrad: estimating gradients for waveform generation. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, Online, 2021. 7

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montreal, Canada, 2018a. 3

Chen, W. Y., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. Stein points. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, Stockholm, Sweden, 2018b. 9

Cheng, K. F. and Chu, C. K. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004. doi: 10.3150/bj/1093265631. 27

Chung, H. and Ye, J. C. Score-based diffusion models for accelerated MRI. *Medical Image Analysis*, 80:102479, 2022. doi: 10.1016/j.media.2022.102479. 2

Corander, J., Fraser, C., Gutmann, M. U., Arnold, B., Hanage, W. P., Bentley, S. D., Lipsitch, M., and Croucher, N. J. Frequency-dependent selection in vaccine-associated pneumococcal population dynamics. *Nature Ecology & Evolution*, 1(12):1950–1960, 2017. doi: 10.1038/s41559-017-0337-x. 1

Cranmer, K., Pavez, J., and Louppe, G. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2016. 6

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117. 1

Dax, M., Wildberger, J., Buchholz, S., Green, S. R., Macke, J. H., and Schölkopf, B. Flow matching for scalable simulation-based inference. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, LA, 2023. 2, 4, 7, 9, 37

De Nicolao, G., Sparacino, G., and Cobelli, C. Nonparametric input estimation in physiological systems: Problems, methods, and case studies. *Automatica*, 33(5):851–870, 1997. doi: 10.1016/S0005-1098(96)00254-3. 34

Deistler, M., Goncalves, P. J., and Macke, J. H. Truncated proposals for scalable and hassle-free simulation-based inference. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, 2022a. 2, 4, 6, 7, 8, 9, 23, 35, 37

Deistler, M., Macke, J. H., and Gonçalves, P. J. Energy-efficient network activity from disparate circuit parameters. *Proceedings of the National Academy of Sciences*, 119(44):e2207632119, 2022b. doi: 10.1073/pnas.2207632119. 8, 37

Delyon, B. and Portier, F. Integral approximation by kernel smoothing. *Bernoulli*, 22(4):2177–2208, 2016. doi: 10.3150/15-BEJ725. 27

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021. 2, 3, 7

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017. 9

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. Reduce, reuse, recycle: compositional generation with energy-based diffusion models and MCMC. In *Proceedings of the 40th International Conference of Machine Learning (ICML 2023)*, Honolulu, HI, 2023. 31, 38

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019. 5

Durkan, C., Murray, I., and Papamakarios, G. On contrastive learning for likelihood-free inference. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online, 2020. 2, 5, 6

Fan, Y., Nott, D. J., and Sisson, S. A. Approximate Bayesian computation via regression density estimation. *Stat*, 2(1): 34–48, 2013. ISSN 2049-1573. doi: 10.1002/sta4.15. 6

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. How to train your neural ODE: the world of Jacobian and kinetic regularization. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online, 2020. 9

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

Frazier, D. T., Nott, D. J., Drovandi, C., and Kohn, R. Bayesian inference using synthetic likelihood: asymptotics and adjustments. *Journal of the American Statistical Association*, pp. 1–12, 2022. doi: 10.1080/01621459.2022.2086132. 6

Fu, H., Yang, Z., Wang, M., and Chen, M. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024. 18

Geffner, T., Papamakarios, G., and Mnih, A. Compositional score modeling for simulation-based inference. In *Proceedings of the 40th International Conference of Machine Learning (ICML 2023)*, Honolulu, HI, 2023. 2, 7, 31, 33, 34

Gelfand, A. E., Smith, A. F. M., and Lee, T.-M. Bayesian analysis of constrained parameter and truncated data problems Using Gibbs sampling. *Journal of the American Statistical Association*, 87(418):523–532, 1992. doi: 10.1080/01621459.1992.10475235. 25

Gelman, A., Carlin, J., Stern, H., and Rubin, D. *Bayesian data analysis*. Chapman and Hall, London, 1995. 5, 24

Glaser, P., Arbel, M., Hromadka, S., Doucet, A., and Gretton, A. Maximum likelihood learning of unnormalized models for simulation-based inference. *arXiv preprint arXiv:2210.14756*, 2022. 2, 8

Glockler, M., Deistler, M., and Macke, J. H. Variational methods for simulation-based inference. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022. 2, 6, 8, 9, 34, 35

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9, 2020. doi: 10.7554/eLife.56261. 1, 8

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, Montreal, Canada, 2014. 1

Gourieroux, C., Monfort, A., and Renault, E. Indirect inference. *Journal of Applied Econometrics*, 8(S1):S85–S118, 1993. doi: 10.1002/jae.3950080507. 1

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, 2019. 3, 9

Greenberg, D. S., Nonnenmacher, M., and Macke, J. H. Automatics posterior transformation for likelihood-free inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, 2019. 1, 5, 6, 7, 29, 30, 32, 34

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Scholkopf, B. Covariate shift by kernel mean matching. In Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. (eds.), *Dataset Shift in Machine Learning*, chapter 8, pp. 131–160. MIT Press, Cambridge MA, 2009. 27

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018. doi: 10.1007/s11222-017-9738-6. 6

Haddad, S. A. and Marder, E. Recordings from the c. borealis stomatogastric nervous system at different temperatures in the decentralized condition, 2021. URL `https://zenodo.org/records/5139650`. 8

Haussmann, U. G. and Pardoux, E. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, oct 1986. doi: 10.1214/aop/1176992362. 3

Henmi, M., Yoshida, R., and Eguchi, S. Importance sampling via the estimated sampler. *Biometrika*, 94(4):985–991, 2007. doi: 10.1093/biomet/asm076. 27

Hermans, J., Begy, V., and Louppe, G. Likelihood-free MCMC with amortized approximate ratio estimators. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online, 2020. 2, 6

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., Begy, V., and Louppe, G. A trust crisis in simulation-based inference? Your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*, 2022. 35

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, Online, 2020. 2, 7

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, 2022. 7

Holden, P. B., Edwards, N. R., Hensman, J., and Wilkinson, R. D. ABC for climate: dealing with expensive simulators. In Sisson, S. A., Fan, Y., and Beaumont, M. A. (eds.), *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, New York, 2018. doi: 10.1201/9781315117195. 1

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866. 9

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. 2

Izbicki, R., Lee, A. B., and Schafer, C. M. High-dimensional density ratio estimation with extensions to approximate likelihood computation. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS 2014)*, Reykjavik, Iceland, 2014. 6

Kanamori, T., Hido, S., and Sugiyama, M. A least squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009. 27

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, 2022. 9

Kingma, D. P. and Ba, J. Adam: a method for stochastic optimisation. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, 2015. 7, 37

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. DiffWave: a versatile diffusion model for audio synthesis. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, Online, 2021. 7

Korba, A., Salim, A., Arbel, M., Luise, G., and Gretton, A. A non-asymptotic analysis for Stein variational gradient descent. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020. 32

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modelling. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, Kigali, Rwanda, 2023. 7

Liu, Q. and Lee, J. D. Black-box importance sampling. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, Fort Lauderdale, FL, 2017. 27

Liu, Q., Lee, J. D., and Jordan, M. A kernelized Stein discrepancy for goodness-of-fit tests. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, New York, NY, 2016. 32

Liu, S., Kanamori, T., Jitkrittum, W., and Chen, Y. Fisher efficient inference of intractable models. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019. 27

Liu, S., Yu, J., Simons, J., Yi, M., and Beaumont, M. Minimizing f-divergences by interpolating velocity fields. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, Vienna, Austria, 2024. 32

Lopez-Paz, D. and Oquab, M. Revisiting classifier two-sample tests. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017. 7

Lotka, A. J. Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, 6(7):410–415, 1920. 34

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, 2022. 9

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. 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. 1, 5, 6, 27, 34

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., and Macke, J. H. Likelihood-free inference with emulator networks. In *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference (AABI 2019)*, volume 96, pp. 32–53, Vancouver, Canada, 2019. 1, 6

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. Benchmarking simulation-based inference. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021)*, Online, 2021. 7, 32, 34

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, dec 2003. doi: 10.1073/pnas.0306899100. 1

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. Unrolled generative adversarial networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017. 2

Miller, B. K., Cole, A., Forré, P., Louppe, G., and Weniger, C. Truncated marginal neural ratio estimation. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021. 2, 6

Mittal, G., Engel, J., Hawthorne, C. G.-M., and Simon, I. Symbolic music generation with diffusion models. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, Online, 2021. 7

Nijkamp, E., Pang, B., Han, T., Zhu, S.-C., and Wu, Y. N. Learning multi-layer latent variable model via variational optimization of short run MCMC for approximate inference. In *European Conference on Computer Vision*, pp. 361–378, Online, 2020. 38

Ong, V. M. H., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. Variational Bayes with synthetic likelihood. *Statistics and Computing*, 28(4):971–988, 2018. doi: 10.1007/s11222-017-9773-3. 6

Papamakarios, G. and Murray, I. 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. 1, 5, 6, 7, 24

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Red Hook, NY, 2017. 1, 9

Papamakarios, G., Sterratt, D. C., and Murray, I. Sequential neural likelihood: fast likelihood-free inference with autoregressive flows. In *22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)*, Okinawa, Japan, 2019. 2, 5, 6, 34

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 1, 27

Pham, K. C., Nott, D. J., and Chaudhuri, S. A note on approximating ABC-MCMC using flexible classifiers. *Stat*, 3(1):218–227, 2014. doi: 10.1002/sta4.56. 6

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. Grad-TTS: a diffusion probabilistic model for text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, Online, 2021. 7

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018. doi: 10.1080/10618600.2017.1302882. 6

Prinz, A. A., Billimoria, C. P., and Marder, E. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. *Journal of neurophysiology*, 90(6):3998–4015, 2003. doi: 10.1152/jn.00641.2003. 8

Prinz, A. A., Bucher, D., and Marder, E. Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7(12):1345–1352, 2004. doi: 10.1038/nn1352. 8

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999. doi: 10.1093/oxfordjournals.molbev.a026091. 1

Qin, J. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998. doi: 10.1093/biomet/85.3.619. 27

Ramesh, P., Lueckmann, J.-M., Boelts, J., Tejero-Cantero, Á., Greenberg, D. S., Gonçalves, P. J., and Macke, J. H. GATSBI: generative adversarial training for simulation-based inference. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022. 2

Ratmann, O., Jørgensen, O., Hinkley, T., Stumpf, M., Richardson, S., and Wiuf, C. Using likelihood-free inference to compare evolutionary dynamics of the protein networks of H. pylori and P. falciparum. *PLOS Computational Biology*, 3(11):e230, 2007. doi: 10.1371/journal.pcbi.0030230. 1

Rubin, D. Comment on 'The calculation of posterior distributions by data augmentation' by Tanner M. and Wong W.H. *Journal of the American Statistical Association*, 82: 543–546, 1987. 5, 24

Rubin, D. Using the SIR algorithm to simulate posterior distributions. In Bernardo, J., Degroot, M., Lindley, D., and Smith, A. (eds.), *Bayesian Statistics 3: Proceedings of the Third Valencia International Meeting*. Oxford University Press, 1988. 5, 24

Salimans, T. and Ho, J. Should EBMs model the energy or the score? In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021): Energy Based Models Workshop*, Online, 2021. 38

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. Improved techniques for training GANs. In *Proceedings of the 30th Conference on Neural Information Processings Systems (NIPS 2016)*, Barcelona, Spain, 2016. 2

Schmitt, M., Pratz, V., Kothe, U., Burkner, P.-C., and Radev, S. T. Consistency models for scalable and fast simulation-based inference. *arXiv preprint arXiv:2312.05440*, 2023. 7

Simons, J., Liu, S., and Beaumont, M. Variational likelihood-free gradient descent. In *Proceedings of the 4th Symposium on Advances in Approximate Bayesian Inference (AABI 2021)*, pp. 1–9, Online, 2021. 6

Sisson, S., Fan, Y., and Beaumont, M. A. Overview of approximate Bayesian computation. In *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC Press., New York, 2018. doi: 10.1201/9781315117195. 1

Sisson, S. A., Fan, Y., and Tanaka, M. M. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007. doi: 10.1073/pnas.0607208104. 34

Skilling, J. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*, pp. 455–466. Springer, Dordrecht, 1989. 9

Smith, A. F. M. and Gelfand, A. E. Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992. doi: 10.1080/00031305.1992.10475856. 5, 24

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Lille, France, 2015. 7

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019. 2, 7, 31

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, Online, 2020. 2, 36

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: a scalable approach to density and score estimation. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI 2020)*, Online, 2020. 2

Song, Y., Sohl-Dickstein, J., Kingma, D., Kumar, A., Ermon, S., and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, Online, 2021. 2, 3, 7, 36

Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, Online, 2022. 7

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *Proceedings of the 40th International Conference of Machine Learning (ICML 2023)*, Honolulu, HI, 2023. 7

Sterratt, D., Graham, B., Gillies, A., and Willshaw, D. *Principles of Computational Modelling in Neuroscience*. Cambridge University Press, Cambridge, 2011. doi: 10.1017/CBO9780511975899. 1

Stoye, M., Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. Likelihood-free inference with an improved cross-entropy estimator. In *Proceedings of the 2nd Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019)*, Vancouver, Canada, 2019. 6

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008. doi: 10.1007/s10463-008-0197-x. 27

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012. ISBN 9780521190176. 27

Tashiro, Y., Song, J., Song, Y., and Ermon, S. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, 2021. 2, 3, 7, 17

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997. doi: 10.1093/genetics/145.2.505. 1

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31, 2022. doi: 10.1214/20-BA1238. 2, 6

Tran, D., Ranganath, R., and Blei, D. M. Hierarchical implicit models and likelihood-free variational inference. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, 2017. 6

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S., and Sugiyama, M. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155, 2009. doi: 10.1137/1.9781611972788.40. 27

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, 2017. 36

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142. 2

Wiqvist, S., Frellsen, J., and Picchini, U. Sequential neural posterior and likelihood approximation. *arXiv preprint arXiv:2102.06522*, 2021. 6

Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010. doi: 10.1038/nature09319. 1, 6

Xiong, Y., Yang, X., Zhang, S., and He, Z. An efficient likelihood-free Bayesian inference method based on sequential neural posterior estimation. *arXiv preprint arXiv:2311.12530*, 2023. 32

Yamada, M. and Sugiyama, M. Direct importance estimation with Gaussian mixture models. *IEICE Transactions on Information and Systems*, E92.D(10):2159–2162, 2009. doi: 10.1587/transinf.E92.D.2159. 27

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, Kigali, Rwanda, 2023. 9

A. Neural Posterior Score Estimation: Theoretical Results

A.1. Derivation of the NPSE Loss Function

In this section we include a self-contained proof that the minimiser of the denoising posterior score matching objective in (7) is equal to the minimiser of the posterior score matching objective in (6). Similar results can also be found in Batzolis et al. (2021); Tashiro et al. (2021).

Our proof begins with the observation that

$$\mathcal{J}_{\text{post}}^{\text{DSM}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||s_\psi(\theta_t,x,t) - \nabla_\theta \log p_t(\theta_t|x)||^2\right]\mathrm{d}t \tag{20}$$

$$= \frac{1}{2}\int_0^T \lambda_t \Bigg[\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||s_\psi(\theta_t,x,t)||^2\right]}_{\Omega_t^1} - 2\underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}[s_\psi^\top(\theta_t,x,t)\nabla_\theta \log p_t(\theta_t|x)]}_{\Omega_t^2} \tag{21}$$

$$+ \underbrace{\mathbb{E}_{(\theta_t,x)\sim p_t(\theta_t,x)}\left[||\nabla_\theta \log p_t(\theta_t|x)||^2\right]}_{\Omega_t^3}\Bigg]\mathrm{d}t.$$

For the first term Ω_t^1, we have that

$$\Omega_t^1 = \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)||s_\psi(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x \tag{22}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t|x)p(x)||s_\psi(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x \tag{23}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\left[\int_{\mathbb{R}^n} p_{t|0}(\theta_t|x,\theta_0)p(\theta_0|x)\mathrm{d}\theta_0\right]p(x)||s_\psi(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}x \tag{24}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p_{t|0}(\theta_t|\theta_0)p(\theta_0|x)p(x)||s_\psi(\theta_t,x,t)||^2\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x \tag{25}$$

$$= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_\psi(\theta_t,x,t)||^2\right]. \tag{26}$$

For the second term Ω_t^2, we have that

$$\Omega_t^2 = \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t,x)s_\psi^\top(\theta_t,x,t)\nabla_\theta \log p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \tag{27}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p_t(\theta_t|x)p(x)s_\psi^\top(\theta_t,x,t)\nabla_\theta \log p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \tag{28}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p(x)s_\psi^\top(\theta_t,x,t)\nabla_\theta p_t(\theta_t|x)\mathrm{d}\theta_t\mathrm{d}x \tag{29}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n} p(x)s_\psi^\top(\theta_t,x,t)\nabla_{\theta_t}\left[\int p_{t|0}(\theta_t|x,\theta_0)p(\theta_0|x)\mathrm{d}\theta_0\right]\mathrm{d}\theta_t\mathrm{d}x \tag{30}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(\theta_0,x)s_\psi^\top(\theta_t,x,t)\nabla_{\theta_t}p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x \tag{31}$$

$$= \int_{\mathbb{R}^d}\int_{\mathbb{R}^n}\int_{\mathbb{R}^n} p(\theta_0,x)p_{t|0}(\theta_t|\theta_0)s_\psi^\top(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_t\mathrm{d}\theta_0\mathrm{d}x \tag{32}$$

$$= \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}[s_\psi^\top(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)]. \tag{33}$$

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

$$\mathcal{J}_{\text{post}}^{\text{DSM}}(\psi) \propto \frac{1}{2}\int_0^t \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_\psi(\theta_t,x,t)||^2 - 2s_\psi^\top(\theta_t,x,t)\nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right]\mathrm{d}t \tag{34}$$

$$\propto \frac{1}{2}\int_0^t \lambda_t \mathbb{E}_{(\theta_0,x)\sim p(\theta_0,x),\theta_t\sim p_{t|0}(\theta_t|\theta_0)}\left[||s_\psi(\theta_t,x,t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)||^2\right]\mathrm{d}t. \tag{35}$$

A.2. Error Bounds for NPSE

We now present error bounds for NPSE in the fully deterministic sampling regime, assuming an L^2 bound on the approximation error and a mild regularity condition on the target posterior distribution. Our results are based on those obtained in Benton et al. (2024).

A.2.1. Notation

We begin by setting up some basic notation. We first recall the definition of the probability flow ODE, now for fixed $x = x_{\text{obs}}$, and $t \in [0,1]$. This ODE is given by

$$\frac{\mathrm{d}\theta_t^{\vartheta}}{\mathrm{d}t} = \underbrace{\left[-f(\theta_t^{\vartheta}, t) + \frac{1}{2}g^2(t)\nabla_\theta \log p_t(\theta_t^{\vartheta}|x_{\text{obs}})\right]}_{v(\theta_t^{\vartheta}, t)}, \quad \theta_0^{\vartheta} = \vartheta \tag{36}$$

for each $\vartheta \in \mathbb{R}^d$. The probability flow ODE defines a deterministic coupling between the reference distribution and the target posterior distribution. In particular, if we define $(\theta_t)_{t\in[0,1]}$ by taking $\vartheta \sim \pi$, and setting $\theta_t = \theta_t^{\varphi}$ for all $t \in [0,1]$, then $\theta_1 \sim p(\cdot|x_{\text{obs}})$. Throughout this section, we will write $v(\theta, t)$ for the velocity field defined by (36), and $v_\psi(\theta, t)$ for the velocity field corresponding to (36) but where the score $\nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ is replaced with its approximation $s_\psi(\theta_t, x_{\text{obs}}, t)$. We suppress notational dependence of these velocity fields on x_{obs}, since x_{obs} is assumed to be fixed.

It is worth noting several modifications between the definition of the probability flow ODE in this appendix, and the definition in the main text. First, we here assume that time is rescaled so that the probability flow ODE runs for $t \in [0,1]$, rather than $t \in [0,T]$. Second, we now consider time to run in the opposite direction. In particular, running forward in time, the probability flow ODE in (36) transforms the reference distribution to the target distribution, rather than the other way round. We adopt this convention to remain consistent with the setup used in Benton et al. (2024).

A.2.2. Assumptions

We impose the following assumptions, which represent analogues of Assumptions 1, 2, 3, and 4' introduced in Benton et al. (2024).

Assumption A1 (Bound on Joint L^2 Approximation Error)**.** The true and approximate scores $\nabla_\theta \log p_t(\theta_t|x)$ and $s_\psi(\theta_t, x, t)$ satisfy $\int_0^1 \mathbb{E}_{p_t(\theta_t, x)}\left[||s_\psi(\theta_t, x, t) - \nabla_\theta \log p_t(\theta_t|x)||^2\right] \mathrm{d}t \leq \varepsilon^2$.

Assumption A1' (Bound on Conditional L^2 Approximation Error)**.** The true and approximate scores $\nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ and $s_\psi(\theta_t, x_{\text{obs}}, t)$ satisfy $\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||s_\psi(\theta_t, x_{\text{obs}}, t) - \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})||^2\right] \mathrm{d}t \leq \varepsilon_{\text{obs}}^2$.

Assumption A2 (Existence and Uniqueness of Smooth Flows)**.** For each $\vartheta \in \mathbb{R}^d$, and $s \in [0,T]$, there exist $(\eta_{s,t}^{\vartheta})_{t\in[s,1]}$ and $(\iota_{s,t}^{\vartheta})_{t\in[s,1]}$ starting in $\eta_{s,s}^{\vartheta} = \vartheta$ and $\iota_{s,s}^{\theta} = \vartheta$ with velocity fields $v_\psi(\varphi, t)$ and $v(\varphi, t)$ respectively. In addition, $\eta_{s,t}^{\vartheta}$ and $\eta_{s,t}^{\vartheta}$ are continuously differentiable in η, s, and t.

Assumption A3 (Regularity of Approximate Score Function)**.** The score-network $s_\psi(\theta, x_{\text{obs}}, t)$ is differentiable in its first and last inputs. In addition, for each $t \in (0,1)$, there exists a constant L_t such that $s_\psi(\theta, x_{\text{obs}}, t)$ is L_t Lipschitz in θ.

Assumption A4 (Regularity of Data Distribution)**.** Let $\theta \sim p(\cdot|x_{\text{obs}})$. Then, for any $\tau \in (0,\infty)$ and $\xi \sim \mathcal{N}(0, \tau^2\mathbf{I})$ independent of θ, there exists $\lambda \geq 1$ such that $||\text{Cov}_{\xi|\theta'=\vartheta}(\xi)||_{\text{op}} \leq \lambda\tau^2$ for all $\vartheta \in \mathbb{R}^d$, where $\theta' = \theta + \xi$.

Assumption A1 is arguably the most natural assumption on the training error since we learn the score-network s_ψ by minimising the denoising posterior score matching objective in (7). This is proportional to the L^2 approximation erro in (6), which appears in the LHS of the bound in Assumption A1. On the other hand, Assumption A1' is required to apply the results in Benton et al. (2024). Below, we provide an additional technical assumption which can be used to translate Assumption A1 into Assumption A1'.

Assumption B1. Let $\mathcal{A}_{\text{obs}}^{\delta} = \{x \in \mathbb{R}^p : ||x - x_{\text{obs}}||_2 < \delta\}$, for $\delta > 0$. There exists $\delta > 0$ such that $\inf_{x\in\mathcal{A}_{\text{obs}}^{\delta}} p(x) > 0$ and $\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}||s_\psi(\theta_t, x_{\text{obs}}, t) - \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})||^2 \mathrm{d}t \leq (C+1)\inf_{x\in\mathcal{A}_{\text{obs}}^{\delta}} \int_0^1 \mathbb{E}_{p_t(\theta_t|x)}||s_\psi(\theta_t, x, t) - \nabla_\theta \log p_t(\theta_t|x)||^2 \mathrm{d}t$ for some $C \geq 0$.

Alternatively, we can just impose Assumption A1' directly. In this case, the results in Benton et al. (2024) can essentially be applied without modification. We refer to, e.g., Fu et al. (2024, Theorem 3.2) for some conditions under which it is possible to obtain a bound of this type.

A.2.3. Auxiliary Results

In order to extend Benton et al. (2024, Theorem 6) to our setting, we will require some simple additional results. We first establish a lemma which will allow us to translate Assumption A1 into Assumption A1'.

Lemma A.1. *Suppose Assumption A1 and Assumption B1 hold. Then Assumption A1' holds.*

Proof. Let $f(x) = \int_0^1 \mathbb{E}_{p_t(\theta_t|x)}||s_\psi(\theta_t,x,t) - \nabla_\theta \log p_t(\theta_t|x)||^2 \mathrm{d}t$. In addition, let $K = [\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x]^{-1}$ and $K_1 = K(1+C)$. We then have

$$f(x_{\text{obs}}) = \frac{1}{\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x}\left[f(x_{\text{obs}})\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x\right] \tag{37}$$

$$= \frac{1}{\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x}\left[\big[\inf_{\mathrm{x}\in\mathcal{A}^{\delta}_{\text{obs}}} f(x)\big]\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x + \big[f(x_{\text{obs}}) - \inf_{\mathrm{x}\in\mathcal{A}^{\delta}_{\text{obs}}} f(x)\big]\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x\right] \tag{38}$$

$$\leq \frac{1}{\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x}\left[\big[\inf_{\mathrm{x}\in\mathcal{A}^{\delta}_{\text{obs}}} f(x)\big]\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x + C\big[\inf_{\mathrm{x}\in\mathcal{A}^{\delta}_{\text{obs}}} f(x)\big]\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x\right] \tag{39}$$

$$\leq \frac{C+1}{\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x}\left[\int_{\mathcal{A}^{\delta}_{\text{obs}}} f(x)p(x)\mathrm{d}x\right] \tag{40}$$

$$\leq \frac{C+1}{\int_{\mathcal{A}^{\delta}_{\text{obs}}} p(x)\mathrm{d}x}\left[\int_{\mathbb{R}^p} f(x)p(x)\mathrm{d}x\right] \tag{41}$$

$$\leq K_1\varepsilon^2 := \varepsilon^2_{\text{obs}}, \tag{42}$$

where in (39) we have used Assumption B1, in (40) we have used elementary properties of the infimum, in (41) we have used the fact that $f(x)\geq 0$ for all x, and in (42) we have used Assumption A1. □

We next establish a very straightforward lemma which will enable us to convert our L^2 bound on the approximate score function (Assumption A1') into an L^2 bound on the corresponding velocity field in the probability flow ODE.

Lemma A.2. *Suppose Assumption A1' holds. Suppose also that $\sup_{t\in[0,1]} g^4(t) < \infty$. Let $v(\theta,t)$ and $v_\psi(\theta,t)$ be the true and approximate velocity fields for the probability flow ODE, as defined in Section A.2.1. Then there exists $\varepsilon_1 > 0$ such that $\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||v_\psi(\theta_t,t) - v(\theta_t,t)||^2\right]\mathrm{d}t \leq \varepsilon_1^2$.*

Proof. Straightforwardly, we have that

$$\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||v_\psi(\theta_t,t) - v(\theta_t,t)||^2\right]\mathrm{d}t \tag{43}$$

$$= \frac{\inf_{t\in[0,1]}\frac{4}{g^4(t)}}{\inf_{t\in[0,1]}\frac{4}{g^4(t)}}\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||v_\psi(\theta_t,t) - v(\theta_t,t)||^2\right]\mathrm{d}t \tag{44}$$

$$\leq \frac{1}{\inf_{t\in[0,1]}\frac{4}{g^4(t)}}\int_0^1 \frac{4}{g^4(t)}\mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||v_\psi(\theta_t,t) - v(\theta_t,t)||^2\right]\mathrm{d}t \tag{45}$$

$$= \frac{1}{4}\sup_{t\in[0,1]}[g^4(t)]\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[\left|\left|\frac{2\left[v_\psi(\theta_t,t) + f(\theta_t,t)\right]}{g^2(t)} - \frac{2\left[v(\theta_t,t) + f(\theta_t,t)\right]}{g^2(t)}\right|\right|^2\right]\mathrm{d}t \tag{46}$$

$$= \frac{1}{4}\sup_{t\in[0,1]}[g^4(t)]\int_0^1 \mathbb{E}_{p_t(\theta_t|x_{\text{obs}})}\left[||s_\psi(\theta_t,x_{\text{obs}},t) - \nabla_\theta\log p_t(\theta_t|x_{\text{obs}})||^2\right] \tag{47}$$

$$\leq \frac{1}{4}\varepsilon^2\sup_{t\in[0,1]}[g^4(t)] := \varepsilon_1^2, \tag{48}$$

where in (45) we have used elementary properties of the infimum, in (47) we have used the definitions of $v(\theta,t)$ and $v_\psi(\theta,t)$, and in (48) we have used Assumption A1'. □

A.3. Main Result

In order to state our main result, we will require some additional definitions. Following Benton et al. (2024, Corollary 2), we first define the quantities $(\beta_t)_{t\in[0,1]}$ and $(\gamma_t)_{t\in[0,1]}$ as

$$\text{VP ODE:} \quad \gamma_t = R\cos\left[\left(\frac{\pi}{2}-\delta\right)t\right], \quad \beta_t = \sin\left[\left(\frac{\pi}{2}-\delta\right)t\right], \tag{49}$$

$$\text{VE ODE:} \quad \gamma_t \ \text{ decreasing}, \qquad\qquad \beta_t = 1. \tag{50}$$

We also define $(K_t)_{t\in[0,1]}$ according to

$$K_t = \lambda\frac{|\dot{\gamma}_t|}{\gamma_t} + \min\left[\lambda\frac{|\dot{\beta}_t|}{\beta_t}, \lambda^{1/2}R\frac{|\dot{\beta}_t|}{\gamma_t}\right]. \tag{51}$$

Finally, we let $\mathcal{V}$ denote the class of functions $v : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ which are K_t-Lipschitz in θ for all $t \in [0,1]$.

We are now ready to state our main result: a bound on the error of NPSE in the deterministic sampling regime in terms of the L^2 approximation error.

Theorem A.3. *Suppose that Assumption A1 and B1 hold, or that Assumption A1' holds. Suppose also that Assumptions A2, A3, and A4 hold. Let $v_\theta \in \mathcal{V}$. Let $\tilde{\pi}_0 = \mathcal{N}(0,\mathbf{I})$, and let $\tilde{\pi}_1$ equal $p(\cdot|x_{\text{obs}})$ plus Gaussian noise with scale $\gamma_1 \ll 1$. Let $(\eta_t)_{t\in[0,T]}$ be a flow starting in $\tilde{\pi}_0$ with velocity field v_ψ, and let $\hat{\pi}_1$ be the distribution of η_1. Then, with ε_1 defined as in Lemma A.2,*

$$\text{VP ODE:} \quad W_2(\hat{\pi}_1, \tilde{\pi}_1) \leq \varepsilon_1\left[\frac{e}{\gamma_1}\right]^\lambda, \tag{52}$$

$$\text{VE ODE:} \quad W_2(\hat{\pi}_1, \tilde{\pi}_1) \leq \varepsilon_1\left[\frac{1}{\gamma_1}\right]^\lambda. \tag{53}$$

Proof. The result follows directly from Benton et al. (2024, Theorem 6), setting the target distribution $\pi := p(\cdot|x_{\text{obs}})$, for some fixed x_{obs}. We note, in particular, that Assumption 1 in Benton et al. (2024) follows from Assumption A1' (or Assumptions A1 and B1 via Lemma A.1) and Lemma A.2. Meanwhile, our Assumptions A2, A3, and A4 correspond directly to Assumptions 2, 3, and 4' in Benton et al. (2024). □

B. Neural Likelihood Score Estimation

B.1. Overview

In this section we outline an alternative method to the one described in Section 2.2 for learning an approximation to the perturbed posterior score $\nabla_\theta \log p_t(\theta_t|x)$. We refer to this approach as Neural Likelihood Score Estimation (NLSE). Our alternative approach is based on the following decomposition of the posterior score, which follows straightforwardly from Bayes' theorem:

$$\nabla_\theta \log p_t(\theta_t|x) = \nabla_\theta \log p_t(x|\theta_t) + \nabla_\theta \log p_t(\theta_t), \tag{54}$$

where $p_t(x|\theta_t) = \int p(x|\theta_0)p_{0|t}(\theta_0|\theta_t)\mathrm{d}\theta_0$ denotes the conditional density of x given θ_t. This decomposition suggests that, rather than directly targeting the score of the posterior, we could instead train a score network $s_{\psi_{\text{lik}}}(\theta_t, x, t) \approx \nabla_\theta \log p_t(x|\theta_t)$ for the score of the perturbed 'likelihood' $p_t(x|\theta_t)$, and then estimate the perturbed posterior score by setting

$$s_{\psi_{\text{post}}}(\theta_t, x, t) = s_{\psi_{\text{lik}}}(\theta_t, x, t) + \nabla_\theta \log p_t(\theta_t). \tag{55}$$

In certain cases, it is possible to compute the perturbed prior $p_t(\theta_t)$ in closed form, and thus obtain the perturbed prior score $\nabla_\theta \log p_t(\theta_t)$ via automatic differentiation (see Appendix B.2.1). In cases where this is not possible, we can instead approximate this term using an additional score network $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_\theta \log p_t(\theta_t)$ (see Appendix B.2.2).

In order to train the time-varying score network $s_{\psi_{\text{lik}}}(\theta_t, x, t)$, it is once again natural to minimise a weighted Fisher divergence, which now reads

$$\mathcal{J}_{\text{lik}}^{\text{SM}}(\psi_{\text{lik}}) := \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_t(\theta_t,x)}\left[\left|\left|s_{\psi_{\text{lik}}}(\theta_t, x, t) - \nabla_\theta \log p_t(x|\theta_t)\right|\right|^2\right]\mathrm{d}t. \tag{56}$$

Similar to (6), we cannot optimise this objective due to the intractable second term. However, by substituting (54), and arguing as in Appendix A.1, one can show that it is equivalent to minimise the denoising likelihood score matching objective function, given by[3]

$$\mathcal{J}_{\text{lik}}^{\text{DSM}}(\psi_{\text{lik}}) := \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(\theta_0,x)}\left[\left|\left|s_{\psi_{\text{lik}}}(\theta_t,x,t) + \nabla_\theta \log p_t(\theta_t) - \nabla_\theta \log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right] \mathrm{d}t. \tag{57}$$

Similar to (7), given a suitable choice for the drift and diffusion coefficients in (2), the scores $\nabla_{\theta_t} \log p_{t|0}(\theta_t|\theta_0)$ can be computed in closed form. We can compute Monte Carlo estimates of (57), and minimise this using standard methods to obtain $s_{\psi_{\text{lik}}}(\theta_t,x,t) \approx \nabla_{\theta_t} \log p_t(x|\theta_t)$.

B.2. Computing or Estimating the Perturbed Prior Score

B.2.1. Computing the Perturbed Prior Score

For certain choices of the prior, and certain choices of the drift and diffusion coefficients in the forward SDE (2), we can obtain the perturbed prior $p_t(\theta_t) = \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)p(\theta_0)\mathrm{d}\theta_0$ in closed form. We can then obtain the score of the perturbed prior $\nabla_\theta \log p_t(\theta_t)$ using automatic differentiation (e.g., Bartholomew-Biggs et al., 2000).

Suppose, for example, that the drift and diffusion coefficients in (2) are given by $f(\theta_t,t) = 0$ and $g(t) = \tau_t$, where $(\tau_t)_{t\in[0,T]}$ is a positive sequence of reals. In this case, we have $p_{t|0}(\theta_t|\theta) = \mathcal{N}(\theta_t|\theta,\tau_t^2\mathbf{I})$, and can obtain $p_t(\theta_t)$ in closed form for the following common choices of prior.

Uniform Prior. Suppose that $p(\theta) = \mathcal{U}(\theta|a,b)$. We can then compute, writing $\Phi(\cdot|\mu,\sigma^2)$ is the CDF of a univariate Gaussian with mean μ and variance σ^2,

$$\begin{aligned} p_t(\theta_t) &= \int_{\mathbb{R}^d} p(\theta_0)p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_0 && (58)\\ &= \frac{1}{\prod_{i=1}^d (b_i - a_i)} \int_{[a_1,b_1]\times\cdots\times[a_d,b_d]} \mathcal{N}(\theta_t|\theta_0,\tau_t^2\mathbf{I})\mathrm{d}\theta_0 && (59)\\ &= \frac{1}{\prod_{i=1}^d (b_i - a_i)} \prod_{i=1}^d \left(\Phi(b_i|\theta_{t,i},\tau_{i,t}^2) - \Phi(a_i|\theta_{t,i},\tau_{i,t}^2)\right). && (60) \end{aligned}$$

Gaussian Mixture Prior. Suppose that $p(\theta) = \sum_{i=1}^n \alpha_i \mathcal{N}(\theta|\mu_i,\Sigma_i)$. Using standard results (e.g., Bishop, 2006, Equation 2.115), we then have that

$$\begin{aligned} p_t(\theta_t) &= \int_{\mathbb{R}^d} p(\theta_0)p_{t|0}(\theta_t|\theta_0)\mathrm{d}\theta_0 && (61)\\ &= \sum_{i=1}^n \alpha_i \int_{\mathbb{R}^d} \mathcal{N}(\theta_0|\mu_i,\Sigma_i)\mathcal{N}(\theta_t|\theta_0,\tau_t^2\mathbf{I})\mathrm{d}\theta_0 && (62)\\ &= \sum_{i=1}^n \alpha_i \mathcal{N}\left(\theta_t|\mu_i,\Sigma_i + \tau_t^2\mathbf{I}\right). && (63) \end{aligned}$$

B.2.2. Estimating the Perturbed Prior Score

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

$$\mathcal{J}_{\text{pri}}(\psi_{\text{pri}}) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p(\theta_0)p_{t|0}(\theta_t|\theta_0)}\left[\left|\left|s_{\psi_{\text{pri}}}(\theta_t,t) - \nabla_\theta \log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]. \tag{64}$$

We summarise this procedure below.

[3] See also Chao et al. (2022, Theorem 1) for a slightly more general version of this result.

Algorithm 2 Prior Score Estimation

```
Input: prior $p(\theta)$, prior sample budget $N$, dataset $\mathcal{D} = \{\}$.
Outputs: $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_{\theta_t} \log p_t(\theta_t)$
for $i = 1 : N$ do
    Sample $\theta_i \sim p(\theta)$.
    Add $\theta_i$ to $\mathcal{D}$.
end for
Learn $s_{\psi_{\text{pri}}}(\theta_t, t) \approx \nabla_{\theta_t} \log p_t(\theta_t)$ by minimising a Monte Carlo estimate of (64) based on $\mathcal{D}$.
Return: $s_{\psi_{\text{pri}}}(\theta_t, t)$
```

B.3. NPSE versus NLSE

A natural question to ask is whether it is preferable to use NLSE or NPSE. In numerical testing, we observed significantly better performance for NPSE relative to NLSE in cases where it was not possible to compute the score of the perturbed prior, and it was thus necessary to approximate this quantity using an additional score network (see Appendix B.2.2).

Meanwhile, in cases where it was possible to compute the perturbed prior analytically (see Appendix B.2.1), we found little empirical difference between NPSE and NLSE. To illustrate this point, we provide results for four benchmark experiments in Figure 5. For NPSE, we report results using both the VE SDE and VP SDE (see Appendix E.3.1) for the forward noising process. For NLSE, we use the VE SDE (see Appendix E.3.1), as this allowed us to easily compute the perturbed prior in closed form (see Appendix B.2.1).

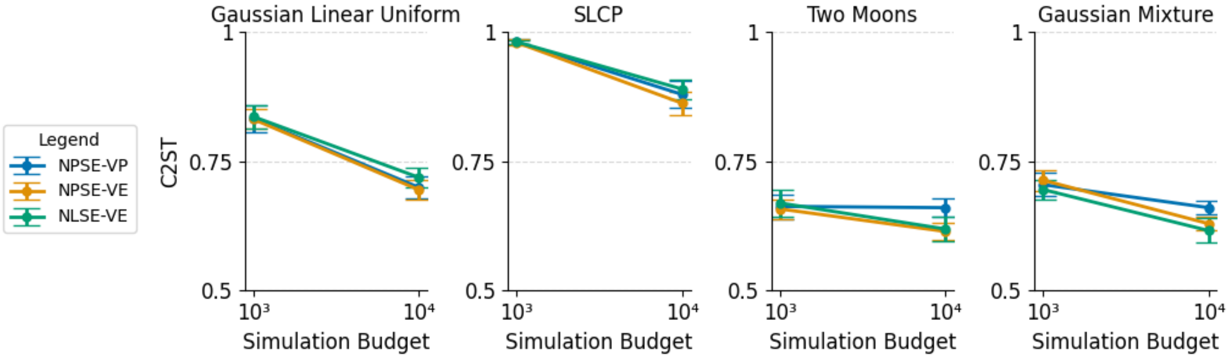


Figure 5. **Comparison between NPSE and NLSE on four benchmark tasks.**

C. Sequential Neural Posterior Score Estimation: Additional Details

In this section we provide further details of the various sequential methods introduced in Section 3: TSNPSE (Section C.1), SNPSE-A (Section C.2), SNPSE-B (Section C.3), and SNPSE-C (Section C.4).

C.1. TSNPSE

We begin with our main sequential algorithm: TSNPSE. In particular, the following section contains a proof of the theoretical result (Proposition 3.1) provided in the main text.

C.1.1. Theoretical Results

Proposition C.1. *Let $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1} \bar{p}^s(\theta)$, where $\bar{p}^0(\theta) = p(\theta)$ and $\bar{p}^s(\theta)$ is defined by (9) for all $s \geq 1$. Suppose that*

$$\Theta_{\text{obs}} \subseteq \text{HPR}_\epsilon(p_\psi^s(\theta|x_{\text{obs}})) \tag{65}$$

for all $s \geq 1$, where $\Theta_{\text{obs}} = \text{supp}(p(\cdot|x_{\text{obs}}))$. Then, writing $\tilde{p}_t^r(\theta_t, x)$ for the distribution of (θ_t, x) when $(\theta_0, x) \sim \tilde{p}^r(\theta, x)$, the minimiser ψ^* of the loss function

$$\mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{SM}}(\psi) := \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{\tilde{p}_t^r(\theta_t,x)}\left[||s_\psi(\theta_t,x,t) - \nabla_\theta \log p_t(\theta_t|x)||^2\right]\mathrm{d}t, \tag{66}$$

or, equivalently, of the loss function

$$\mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{DSM}}(\psi) := \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)}\left[||s_\psi(\theta_t,x,t) - \nabla_\theta \log p_{t|0}(\theta_t|\theta_0)||^2\right]\mathrm{d}t, \tag{67}$$

satisfies $s_{\psi^\star}(\theta_t, x_{\text{obs}}, t) = \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$.

Proof. By definition, we have $\tilde{p}^r(\theta) \propto c^r(\theta)p(\theta)$, where $c^r(\theta) := \frac{1}{r}\sum_{s=0}^{r-1}\mathbb{I}\{\theta \in \Theta^s\}$, with $\Theta^0 = \text{supp}(p(\theta))$ and $\Theta^s = \mathbb{I}\{\theta \in \text{HPR}_\varepsilon(p_\psi^s(\theta|x_{\text{obs}}))\}$ for all $s \geq 1$. Thus, in particular, we can write

$$p^r(\theta) = \frac{c^r(\theta)p(\theta)}{Z^r} := f^r(\theta)p(\theta), \tag{68}$$

where $f^r(\theta) = \frac{c^r(\theta)}{Z^r}$, and where Z^r is the normalisation constant

$$Z^r = \int_{\mathbb{R}^d} c^r(\theta)p(\theta)\mathrm{d}\theta = \frac{1}{r}\sum_{s=0}^{r-1}\int_{\Theta^s} p(\theta)\mathrm{d}\theta. \tag{69}$$

By definition, $c^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1}\mathbb{I}\{\theta \in \Theta^s\} = 1$ for all $\theta \in \cup_{s=0}^{r-1}\Theta^s$. Under the assumption on Θ_{obs}, we thus also have that $c^r(\theta) = 1$ for all $\theta \in \Theta_{\text{obs}}$. It follows that

$$f^r(\theta) = \frac{1}{Z^r} = \text{constant} = A_r \tag{70}$$

for all $\theta \in \Theta_{\text{obs}}$. Thus, in particular, we have that $p^r(\theta) = A_r \cdot p(\theta)$ for all $\theta \in \Theta_{\text{obs}}$; see also Deistler et al. (2022a, Section 6.2).

Now, using standard results (e.g., Batzolis et al., 2021, Theorem 1), we know that $\psi^\star = \arg\min \mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{SM}}(\psi) = \arg\min \mathcal{J}_{\text{post}}^{\text{TSNPSE}-\text{DSM}}(\psi)$ satisfies

$$s_{\psi^\star}(\theta_t, x, t) = \nabla_\theta \log \tilde{p}_t^r(\theta_t|x), \tag{71}$$

where, similar to before, $\tilde{p}_t^r(\theta_t|x) = \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0|x)\mathrm{d}\theta_0$, and $\tilde{p}^r(\theta_0|x) = \frac{\tilde{p}^r(\theta_0)p(x|\theta_0)}{\tilde{p}^r(x)}$. Thus, at the observation x_{obs}, we have that

$$\begin{aligned}
\tilde{p}_t^r(\theta_t|x_{\text{obs}}) &= \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0|x_{\text{obs}})\mathrm{d}\theta_0 && (72)\\
&= \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\frac{\tilde{p}^r(\theta_0)p(x_{\text{obs}}|\theta_0)}{\tilde{p}^r(x_{\text{obs}})}\mathrm{d}\theta_0 && (73)\\
&= \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\frac{f^r(\theta_0)p(\theta_0)p(x_{\text{obs}}|\theta_0)}{\tilde{p}^r(x_{\text{obs}})}\mathrm{d}\theta_0 && (74)\\
&= \int_{\mathbb{R}^d} p_{t|0}(\theta_t|\theta_0)\frac{f^r(\theta_0)p(\theta_0|x_{\text{obs}})p(x_{\text{obs}})}{\tilde{p}^r(x_{\text{obs}})}\mathrm{d}\theta_0 && (75)\\
&\propto \int_{\mathbb{R}^d} f^r(\theta_0)p_{t|0}(\theta_t|\theta_0)p(\theta_0|x_{\text{obs}})\mathrm{d}\theta_0 && (76)\\
&= A_r \cdot \int_{\Theta_{\text{obs}}} p_{t|0}(\theta_t|\theta_0)p(\theta_0|x_{\text{obs}})\mathrm{d}\theta_0 && (77)\\
&= A_r \cdot p_t(\theta_t|x_{\text{obs}}), && (78)
\end{aligned}$$

where the penultimate equality holds since $f^r(\theta) = A_r$ for all $\theta \in \Theta_{\text{obs}}$, and $p(\theta|x_{\text{obs}}) = 0$ for all $\theta \in \mathbb{R}^d \setminus \Theta_{\text{obs}}$. Thus, combining (71) and the logarithmic derivative of (78), we conclude that $s_{\psi^\star}(\theta_t, x_{\text{obs}}, t) = \nabla_\theta \log \tilde{p}_t^r(\theta_t|x_{\text{obs}}) = \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ as required. □

C.2. SNPSE-A

We now provide more details on SNPSE-A, so-named due to its connections with SNPE-A (Papamakarios & Murray, 2016). This algorithm is summarised in Algorithm 3.

C.2.1. Overview

The main steps involved in SNPSE-A can be summarised as follows. For $r = 1$, sample parameters from the prior $\{\theta^1_{0,i}\}_{i=1}^M \sim p(\theta) := p^0_\psi(\cdot|x_{\mathrm{obs}})$. Then, for $r = 1, 2,$

(i) Simulate new data $\{x^r_i\}_{i=1}^M \sim p(\cdot|\theta^r_{0,i})$. Concatenate samples $\{(\theta^r_{0,i}, x^r_i)\}_{i=1}^M$ with those from previous rounds to form $\{(\theta_{0,i}, x_i)\}_{i=1}^{rM} := \bigcup_{s=1}^r \{(\theta^r_{0,i}, x^r_i)\}_{i=1}^M \sim \tilde{p}^r(\theta)p(x|\theta)$, where $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1} p^s_\psi(\theta|x_{\mathrm{obs}})$. Draw times $\{t_i\}_{i=1}^{rM} \sim \mathcal{U}(0,T)$, and samples $\{\theta_{t_i,i}\}_{i=1}^{rM} \sim p_{t|0}(\cdot|\theta_{0,i})$.

(ii) Using these samples, train a time-varying score network $\tilde{s}^r_\psi(\theta_t, x, t)$ to approximate the score of the proposal posterior $\nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$, by minimising a Monte Carlo estimate of the original denoising posterior score matching objective, but now over samples from the proposal prior. That is,

$$\mathcal{J}^{\mathrm{DSM-A}}_{\mathrm{post}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)}\left[\left|\left|\tilde{s}^r_\psi(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]\mathrm{d}t. \tag{79}$$

(iii) Draw samples $\{\theta^{r+1}_{T,i}\}_{i=1}^{M'} \sim \pi(\theta)$, where $M' \geq M$. Simulate the backward SDE (3) or the probability flow ODE (4), substituting $\tilde{s}^r_\psi(\theta_t, x_{\mathrm{obs}}, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x_{\mathrm{obs}})$, to obtain samples $\{\tilde{\theta}^{r+1}_{0,i}\}_{i=1}^{M'} \sim \tilde{p}^r_\psi(\cdot|x_{\mathrm{obs}}) \approx \tilde{p}^r(\cdot|x_{\mathrm{obs}})$.

(iv) Use (approximate) sampling-importance-resampling (SIR) (Rubin, 1987; 1988; Smith & Gelfand, 1992; Gelman et al., 1995) to recover samples $\{\theta^{r+1}_{0,i}\}_{i=1}^M \sim p^r_\psi(\cdot|x_{\mathrm{obs}}) \approx p(\cdot|x_{\mathrm{obs}})$. In particular, draw samples $\{\theta^{r+1}_{0,i}\}_{i=1}^M$ with or without replacement from $\{\tilde{\theta}^{r+1}_{0,i}\}_{i=1}^{M'}$ using sample probabilities w^r_i proportional to

$$h^r_i = \frac{p(\tilde{\theta}^{r+1}_i)}{\tilde{p}^r(\tilde{\theta}^{r+1}_i)}\ , \quad i \in [M']. \tag{80}$$

This corresponds, at the level of densities, to updating the current posterior density estimate as (see also Papamakarios & Murray, 2016)

$$p^r_\psi(\theta|x_{\mathrm{obs}}) \propto \frac{p(\theta)}{\tilde{p}^r(\theta)}\tilde{p}^r_\psi(\theta|x_{\mathrm{obs}}). \tag{81}$$

Algorithm 3 SNPSE-A

```
**Inputs:** Observation $x_{\mathrm{obs}}$, prior $p(\theta) =: p^0_\psi(\theta|x_{\mathrm{obs}})$, simulator $p(x|\theta)$, simulation budget $N$, number of rounds $R$,
(simulations-per-round $M = N/R$)
**Outputs:** Samples $\theta \sim p^r_\psi(\theta|x_{\mathrm{obs}}) \approx p(\theta|x_{\mathrm{obs}})$
for $r = 1, 2$ do
    for $i = 1, \ldots, M$ do
        Draw $\theta_i \sim p^{r-1}_\psi(\theta|x_{\mathrm{obs}})$ using (81) (requires importance weights, see Appendix C.2.3 for details), $x_i \sim p(x|\theta_i)$.
        Add $(\theta_i, x_i)$ to $\mathcal{D}$.
    end for
    Learn $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$ by minimising a Monte Carlo estimate of (79) based on dataset $\mathcal{D}$.
    Get $\tilde{p}^r_\psi(\theta|x_{\mathrm{obs}})$ sampler by substituting $\tilde{s}^r_\psi(\theta_t, x_{\mathrm{obs}}, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x_{\mathrm{obs}})$ into (3) or (4).
end for
**Return:** $\theta \sim p^r_\psi(\theta|x_{\mathrm{obs}})$ using (81) (requires importance weights, see Appendix C.2.3 for details).
```

C.2.2. Theoretical Justification

We can formally justify this procedure as follows. First, using standard results on conditional denoising score matching (e.g., Batzolis et al., 2021), the minimiser

$$\psi^* = \mathrm{argmin}_\psi \mathcal{J}^{\mathrm{DSM-A}}_{\mathrm{post}}(\psi) \tag{82}$$

is such that $\tilde{s}^r_{\psi^*}(\theta_t, x, t) = \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$ for almost all $\theta_t \in \mathbb{R}^d$, $x \in \mathbb{R}^p$, and $t \in [0, T]$. Thus, by substituting the score network $\tilde{s}^r_{\psi^*}(\theta_t, x, t)$ into (3) or (4) we can, in principle, generate samples from the true proposal posterior,

$$\{\tilde{\theta}_i^{r+1}\}_{i=1}^{M'} \sim \tilde{p}^r(\theta|x_{\text{obs}}). \tag{83}$$

It follows, using classical results on SIR (e.g., Gelfand et al., 1992), that if we resample $\{\theta_i^{r+1}\}_{i=1}^{M}$ from $\{\tilde{\theta}_i^{r+1}\}_{i=1}^{M'}$ with or without replacement, using sample probabilities w_i^r proportional to the importance weights

$$h_i^r = \frac{p(\tilde{\theta}_i^{r+1}|x_{\text{obs}})}{\tilde{p}^r(\tilde{\theta}_i^{r+1}|x_{\text{obs}})} \propto \frac{p(\tilde{\theta}_i^{r+1})}{\tilde{p}^r(\tilde{\theta}_i^{r+1})} \tag{84}$$

then, in the limit as $M' \to \infty$, the resulting samples will correspond to i.i.d. draws from the target posterior $p(\cdot|x_{\text{obs}})$, as required. These are precisely the importance weights that we use in SNPSE-A, c.f. (80).

In practice, of course, we will never obtain the minimiser $\psi^* = \text{argmin}_\psi \mathcal{J}_{\text{post}}^{\text{DSM-A}}(\psi)$ but instead some ψ such that, hopefully, $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$ or, alternatively, such that $\tilde{p}^r_\psi(\theta|x_{\text{obs}}) \approx \tilde{p}^r(\theta|x_{\text{obs}})$. Here, as before, we use $\tilde{p}^r_\psi(\theta|x_{\text{obs}})$ to denote the approximation of the true proposal posterior $\tilde{p}^r(\theta|x_{\text{obs}})$ obtained by substituting $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t, x, t)$ into the probability flow ODE (4). Using the score network $\tilde{s}^r_\psi(\theta_t, x, t)$, we can now generate samples from an approximation of the proposal posterior, rather than the true proposal posterior:

$$\{\tilde{\theta}_i^{r+1}\}_{i=1}^{M'} \sim \tilde{p}^r_\psi(\cdot|x_{\text{obs}}) \approx \tilde{p}^r(\theta|x_{\text{obs}}). \tag{85}$$

It follows, once again appealing to standard results on SIR, that in this case the correct probabilities to use in order to recover samples from the true posterior are sample probabilities $\tilde{w}_i^r$ proportional to the importance weights

$$\tilde{h}_i^r = \frac{p(\tilde{\theta}_i^{r+1}|x_{\text{obs}})}{\tilde{p}^r_\psi(\tilde{\theta}_i^{r+1}|x_{\text{obs}})}. \tag{86}$$

These importance weights are only approximately equal to (84), i.e., the importance weights that we actually use in SNPSE-A:

$$\tilde{h}_i^r = \frac{p(\tilde{\theta}_i^{r+1}|x_{\text{obs}})}{\tilde{p}^r_\psi(\tilde{\theta}_i^{r+1}|x_{\text{obs}})} \approx \frac{p(\tilde{\theta}_i^{r+1}|x_{\text{obs}})}{\tilde{p}^r(\tilde{\theta}_i^{r+1}|x_{\text{obs}})} = h_i^r, \tag{87}$$

since we will only ever learn an approximation of the true proposal posterior scores, $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t, x, t)$, and thus an approximation of the true proposal posterior, $\tilde{p}^r_\psi(\theta|x_{\text{obs}}) \approx \tilde{p}^r(\theta|x_{\text{obs}})$, when we minimise the SNPSE-A score-matching objective $\mathcal{J}_{\text{post}}^{\text{DSM-A}}(\psi)$ over a finite number of samples.

It follows that, when we perform a post-hoc correction in the r^{th} round using sample probabilities proportional to h_i^r rather than $\tilde{h}_i^r$, we necessarily introduce an additional approximation into the sequential procedure. This approximation is directly related to the scale of the mismatch between the true proposal posterior $\tilde{p}^r(\theta|x_{\text{obs}})$, and the approximate proposal posterior $\tilde{p}^r_\psi(\theta|x_{\text{obs}})$ learned in the r^{th} round.

C.2.3. Computing the Importance Weights

SNPSE-A relies on being able to compute the importance weights in (80). In particular, it is necessary to compute the density ratio between the prior and the proposal prior, viz

$$\frac{p(\theta)}{\tilde{p}^r(\theta)} = \frac{p(\theta)}{\frac{1}{r}\sum_{s=0}^{r-1} p^s_\psi(\theta|x_{\text{obs}})}. \tag{88}$$

Since the prior $p(\theta)$ is typically available in closed form, it just remains to compute the proposal prior $\tilde{p}^r(\theta)$. In the following sections, we outline several possible ways to compute or approxiate this term.

Computing the Proposal Prior. The first and most direct approach is to compute the proposal prior using the probability flow ODE (4) and the instantaneous change-of-variables formula (5). To be precise, by substituting our estimate of the proposal posterior score, $\tilde{s}^r_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x_{\text{obs}})$ into the probability flow ODE (4), it is possible to evaluate the approximate proposal posterior density $\tilde{p}^r_\psi(\theta|x_{\text{obs}}) \approx \tilde{p}^r(\theta|x_{\text{obs}})$ via the instantaneous change of variable formula (5).

Unfortunately, even with access to approximate proposal posterior densities $\tilde{p}_{\psi}^{r}(\theta|x_{\text{obs}})$, it turns out that we can only compute the approximate posterior densities $p_{\psi}^{r}(\theta|x_{\text{obs}})$ for $r = 0, 1$. Thus, by definition, we can only compute the proposal prior $\tilde{p}^{r}(\theta)$ and the importance weights $\frac{p(\theta)}{\tilde{p}^{r}(\theta)}$ for $r = 1, 2$. In other words, it is only possible to compute the required weights in (88) for up to 2 rounds.

To illustrate this, we can consider explicitly the quantities required to compute the proposal priors $\tilde{p}^{1}(\theta), \tilde{p}^{2}(\theta), \ldots$ in rounds $r = 1, 2, 3$. Recall that the proposal priors are defined as the mixture of the posterior estimates $p_{\psi}^{0}(\theta|x_{\text{obs}}), p_{\psi}^{1}(\theta|x_{\text{obs}}), \ldots$ from the previous rounds, c.f. (88). For these proposals to be computable, we must be able to express them in terms of the prior $p(\theta)$, which we assume is known, and the proposal posterior estimates $\tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}}), \tilde{p}_{\psi}^{2}(\theta|x_{\text{obs}}), \ldots$, which we can always access using the instantaneous change of variables formula.

At the start of the 1st round, the current 'posterior estimate' is initialised equal to the prior: $p_{\psi}^{0}(\theta|x_{\text{obs}}) = p(\theta)$. Thus, in the 1st round, the proposal prior is just equal to the prior:

$$\tilde{p}^{1}(\theta) = p_{\psi}^{0}(\theta|x_{\text{obs}}) = p(\theta). \tag{89}$$

At the start of the 2nd round, the current posterior estimate is equal to the proposal posterior estimate obtained in the 1st round: $p_{\psi}^{1}(\theta|x_{\text{obs}}) = \tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}})$. This is because the proposal prior in the 1st round is equal to the prior, and thus the proposal posterior in the 1st round coincides with the posterior. Thus, in the 2nd round, the proposal prior is given by a mixture of the prior and the proposal posterior estimate obtained in the 1st round:

$$\begin{aligned} \tilde{p}^{2}(\theta) &= \frac{1}{2}\left[p_{\psi}^{0}(\theta|x_{\text{obs}}) + p_{\psi}^{1}(\theta|x_{\text{obs}})\right] && (90)\\ &= \frac{1}{2}\left[p(\theta) + \tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}})\right]. && (91) \end{aligned}$$

At the start of 3rd round, the current posterior estimate is equal to the proposal posterior estimate obtained in the 2nd round, reweighted by the ratio between the prior and the proposal prior in the 2nd round:

$$p_{\psi}^{2}(\theta|x_{\text{obs}}) = \frac{1}{Z_{\psi}^{2}}\frac{p(\theta)}{\tilde{p}^{2}(\theta)}\tilde{p}_{\psi}^{2}(\theta|x_{\text{obs}}), \tag{92}$$

where Z_{ψ}^{2} is an (intractable) normalising constant given by $Z_{\psi}^{2} = \tilde{p}^{2}(x_{\text{obs}})/p(x_{\text{obs}})$, where $p(x_{\text{obs}}) = \int_{\mathbb{R}^d} p(\theta)p(x_{\text{obs}}|\theta)\mathrm{d}\theta$ and $\tilde{p}^{2}(x_{\text{obs}}) = \int_{\mathbb{R}^d} \tilde{p}^{2}(\theta)p(x_{\text{obs}}|\theta)\mathrm{d}\theta$. Thus, in the 3rd round, the proposal prior is given by a mixture of the prior, the proposal posterior estimate obtained in the 1st round, and the posterior estimate obtained in the 2nd round:

$$\begin{aligned} \tilde{p}^{3}(\theta) &= \frac{1}{3}\left[p_{\psi}^{0}(\theta|x_{\text{obs}}) + p_{\psi}^{1}(\theta|x_{\text{obs}}) + p_{\psi}^{2}(\theta|x_{\text{obs}})\right] && (93)\\ &= \frac{1}{3}\left[p(\theta) + \tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}}) + \frac{1}{Z_{\psi}^{2}}\frac{p(\theta)}{\frac{1}{2}\left[p(\theta) + \tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}})\right]}\tilde{p}_{\psi}^{2}(\theta|x_{\text{obs}})\right]. && (94) \end{aligned}$$

Crucially, this proposal not only depends on the prior $p(\theta)$ and the proposal posterior estimates $\tilde{p}_{\psi}^{1}(\theta|x_{\text{obs}}), \tilde{p}_{\psi}^{2}(\theta|x_{\text{obs}}), \ldots$, but also on an intractable normalising constant Z_{ψ}^{2}. Thus, without additional approximations, we cannot use this approach in the 3rd round, or any future rounds.

Approximating the Proposal Prior. An alternative approach is to approximate the proposal prior, or each component of the proposal prior, using samples. In this case, we replace the proposal prior $\tilde{p}^{r}(\theta)$ in (88) by an approximate proposal prior $\hat{p}^{r}(\theta) \approx \tilde{p}^{r}(\theta)$, which we obtain using samples $\theta \sim \tilde{p}^{r}(\theta)$. The advantages of this approach are that (a) it can be applied if we use more than 2 rounds and (b) we no longer need to use the probability flow ODE (4) and the instantaneous change-of-variables formula (5) to compute densities.

To see this, let us once again consider the quantities (i.e., the proposal priors) required to compute the importance weights in rounds $r = 1, 2, 3$. At the start of the 1st round, the current 'posterior estimate' is once again defined to be equal to the prior. Thus, the proposal prior $\tilde{p}^{1}(\theta)$ is equal to the prior $p(\theta)$, as in (89). In this case, we can just set the approximate proposal prior equal to the original proposal prior:

$$\hat{p}^{1}(\theta) := \tilde{p}^{1}(\theta). \tag{95}$$

At the start of the 2^{nd} round, the current posterior estimate is equal to the proposal posterior estimate from the 1^{st} round, as argued after (89). Thus, as before, the proposal prior is equal to a mixture of the prior $p(\theta)$ and the proposal posterior estimate $\tilde{p}^1_\psi(\theta|x_{\text{obs}})$ from the 1^{st} round:

$$\tilde{p}^2(\theta) = \frac{1}{2}[p(\theta) + \tilde{p}^1_\psi(\theta|x_{\text{obs}})] \tag{96}$$

Unlike before, suppose now that we can generate samples from $\tilde{p}^1_\psi(\theta|x_{\text{obs}})$ and thus from $\tilde{p}^2(\theta) = \frac{1}{2}[p(\theta) + \tilde{p}^1_\psi(\theta|x_{\text{obs}})]$, but that we cannot explicitly evaluate $\tilde{p}^1_\psi(\theta|x_{\text{obs}})$, nor $\tilde{p}^2(\theta) = \frac{1}{2}[p(\theta) + \tilde{p}^1_\psi(\theta|x_{\text{obs}})]$. For example, it may be too expensive to solve the instantaneous change-of-variables formula (5) to evaluate these densities. We then have two natural options for approximating $\tilde{p}^2(\theta)$. The first is to directly approximate $\hat{p}^2(\theta) \approx \tilde{p}^2(\theta)$ using samples $\theta \sim \tilde{p}^2(\theta)$. The second is to approximate $\hat{p}^1_\psi(\theta|x_{\text{obs}}) \approx \tilde{p}^1_\psi(\theta|x_{\text{obs}})$ using samples $\theta \sim \tilde{p}^1_\psi(\theta|x_{\text{obs}})$, and then to approximate $\tilde{p}^2(\theta)$ using

$$\hat{p}^2(\theta) := \frac{1}{2}\left[p(\theta) + \hat{p}^1_\psi(\theta|x_{\text{obs}})\right]. \tag{97}$$

At the start of the 3^{rd} round, the current posterior estimate is equal to the approximate proposal posterior estimate from the 2^{nd} round, reweighted by the ratio between the prior and the approximate proposal prior from the 2^{nd} round:

$$p^2_\psi(\theta|x_{\text{obs}}) = \frac{1}{\hat{Z}^2_\psi}\frac{p(\theta)}{\hat{p}^2(\theta)}\hat{p}^2_\psi(\theta|x_{\text{obs}}), \tag{98}$$

where $\hat{p}^2_\psi(\theta|x_{\text{obs}})$ is the approximation of the proposal posterior $\hat{p}^2(\theta|x_{\text{obs}}) \propto \hat{p}^2(\theta)p(x_{\text{obs}}|\theta)$ associated with the approximate proposal prior $\hat{p}^2(\theta)$, and $\hat{Z}^2_\psi$ is the appropriate normalising constant. Explicitly, we now have $\hat{Z}^2_\psi = \frac{\hat{p}^2(x_{\text{obs}})}{p(x_{\text{obs}})}$, where $p(x_{\text{obs}}) = \int_{\mathbb{R}^d} p(\theta)p(x_{\text{obs}}|\theta)\mathrm{d}\theta$ and $\hat{p}^2(x_{\text{obs}}) = \int_{\mathbb{R}^d} \hat{p}^2(\theta)p(x_{\text{obs}}|\theta)\mathrm{d}\theta$. Thus, in the 3^{rd} round, the proposal prior $\tilde{p}^3(\theta)$ is equal to a mixture of the prior, the approximate proposal posterior estimate obtained in 1^{st} round, and the approximate posterior estimate in (98) obtained in the 2^{nd} round. That is, the mixture defined in (94), but now with $\tilde{p}(\cdot)$ replaced everywhere by $\hat{p}(\cdot)$.

We cannot directly evaluate $p^2_\psi(\theta|x_{\text{obs}})$ in (98) due to the intractable normalising constant $\hat{Z}^2_\psi$. We can, however, still generate samples $\theta \sim p^2_\psi(\theta|x_{\text{obs}})$ using, e.g., SIR, since this only requires that we can sample from $\hat{p}^2_\psi(\theta|x_{\text{obs}})$, and that we can evaluate both $p(\theta)$ and $\hat{p}^2(\theta)$. Thus, by construction, we can also generate samples from the proposal prior, $\theta \sim \tilde{p}^3(\theta)$. Similar to before, we can then approximate $\tilde{p}^3(\theta)$ using samples, either directly or by estimating the mixture components individually.

For subsequent rounds, we can proceed in precisely the same fashion. In particular, first generate samples from the current estimate of the proposal posterior. Then use SIR to obtain samples from the current posterior estimate. Finally, use these samples to approximate the current posterior estimate, and use this approximation to approximate the next proposal prior.

The disadvantage of this approach is that it necessitates additional approximations in each round. This being said, perhaps somewhat surprisingly, in other contexts the use of approximate importance weights rather than exact importance weights can actually improve performance (Henmi et al., 2007; Delyon & Portier, 2016); see also the discussion in Liu & Lee (2017). Regarding the proposal prior approximations, there are several possibilities: e.g., a kernel density estimator (Delyon & Portier, 2016), or a normalising flow (Papamakarios et al., 2021). We leave a more thorough investigation of this approach to future work.

Approximating the Importance Weights. One final option is to approximate the density ratio in (88) using samples, rather than just approximating the proposal prior. This is the subject of (two sample) density ratio estimation (DRE) (Sugiyama et al., 2012). There are various approaches to two-sample DRE, amongst others, moment matching (Gretton et al., 2009), probabilistic classification (Qin, 1998; Cheng & Chu, 2004; Bickel et al., 2007), and ratio matching (Sugiyama et al., 2008; Kanamori et al., 2009; Tsuboi et al., 2009; Yamada & Sugiyama, 2009). In our case, not only do we have access to samples from the prior, but we can also evaluate the density. In this setting, Stein density ratio estimation (SDRE) provides an alternative approach (Liu et al., 2019). Once again, we leave a more detailed investigation into this approach to future work.

C.3. SNPSE-B

Next, we provide additional details on SNPSE-B, which can be seen as the score-based analogue of SNPE-B (Lueckmann et al., 2017). This algorithm is summarised in Algorithm 4.

C.3.1. Overview

The main steps involved in SNPSE-B are as follows. For $r = 1$, sample parameters from the prior $\{\theta_{0,i}^1\}_{i=1}^M \sim p(\theta) := p_\psi^0(\cdot|x_{\text{obs}})$. Then, for all $r \geq 1$,

(i) Simulate new data $\{x_i^r\}_{i=1}^M \sim p(\cdot|\theta_{0,i}^r)$. Concatenate samples $\{(\theta_{0,i}^r, x_i^r)\}_{i=1}^M$ with those from previous rounds to form $\{(\theta_{0,i}, x_i)\}_{i=1}^{rM} := \bigcup_{s=1}^r \{(\theta_{0,i}^r, x_i^r)\}_{i=1}^M \sim \tilde{p}^r(\theta)p(x|\theta)$, where $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1} p_\psi^s(\theta|x_{\text{obs}})$. Draw times $\{t_i\}_{i=1}^{rM} \sim \mathcal{U}(0,T)$, and samples $\{\theta_{t_i,i}\}_{i=1}^{rM} \sim p_{t|0}(\cdot|\theta_{0,i})$.

(ii) Using these samples, train a time-varying score network $s_\psi(\theta_t, x, t)$ to approximate the score of the posterior $\nabla_\theta \log p_t(\theta_t|x)$, by minimising a Monte Carlo estimate of

$$\mathcal{J}_{\text{post}}^{\text{DSM-B}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)}\left[\frac{p(\theta_0)}{\tilde{p}^r(\theta_0)}\left|\left|s_\psi(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]\mathrm{d}t. \tag{99}$$

(iii) Draw samples $\{\theta_{T,i}^{r+1}\}_{i=1}^M \sim \pi(\theta)$. Simulate the backward SDE (3) or the probability flow ODE (4), substituting $s_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$, to obtain samples $\{\theta_{0,i}^{r+1}\}_{i=1}^M \sim p_\psi^r(\cdot|x_{\text{obs}}) \approx p(\cdot|x_{\text{obs}})$.

Algorithm 4 SNPSE-B

```
Inputs: Observation $x_{\text{obs}}$, prior $p(\theta) =: p_\psi^0(\theta|x_{\text{obs}})$, simulator $p(x|\theta)$, simulation budget $N$, number of rounds $R$,
(simulations-per-round $M = N/R$)
Outputs: $p_\psi(\theta|x_{\text{obs}}) \approx p(\theta|x_{\text{obs}})$
for $r = 1, \ldots, R$ do
  for $i = 1, \ldots, M$ do
    Draw $\theta_i \sim p_\psi^{r-1}(\theta|x_{\text{obs}})$, $x_i \sim p(x|\theta_i)$
    Add $(\theta_i, x_i)$ to $\mathcal{D}$
  end for
  Learn $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ by minimising a Monte Carlo estimate of (99) based on dataset $\mathcal{D}$ (requires
  importance weights, see Appendix C.3.3 for details).
  Get $p_\psi^r(\theta|x_{\text{obs}})$ sampler by substituting $s_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ into (3) or (4).
end for
Return: $p_\psi^R(\theta|x_{\text{obs}})$
```

C.3.2. Theoretical Justification

The theoretical justification for SNPSE-B is rather straightforward. In particular, observe that

$$\mathcal{J}_{\text{post}}^{\text{DSM-B}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)}\left[\frac{p(\theta_0)}{\tilde{p}^r(\theta_0)}\left|\left|s_\psi(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]\mathrm{d}t \tag{100}$$

$$= \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)p(\theta_0)}\left[\left|\left|s_\psi(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]\mathrm{d}t. \tag{101}$$

This is nothing more than the original denoising posterior score matching objective in (7), which we know is minimised by ψ^* such that $s_{\psi^*}(\theta_t, x, t) = \nabla_\theta \log p_t(\theta_t|x)$ (e.g., Batzolis et al., 2021). Thus, substituting $s_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ into the backward SDE (3) or the probability flow ODE (4), will indeed result in samples approximately distributed according to $p(\theta|x_{\text{obs}})$, with no need for an additional correction.

C.3.3. Computing the Importance Weights

Similar to SNPSE-A, SNPSE-B relies on our ability to compute or approximate the importance weights $p(\theta)/\tilde{p}^r(\theta)$, which now appear in the objective function in (99). We refer to Appendix C.2.3 for a detailed discussion of how to compute or approximate these weights.

C.4. SNPSE-C

Finally, we provide additional details regarding SNPSE-C, which can be seen as the score-based analogue of SNPE-C (Greenberg et al., 2019). This algorithm is summarised in Algorithm 5.

C.4.1. Overview

The main components of the SNPSE-C algorithm are summarised below. For $r = 1$, sample parameters from the prior $\{\theta_{0,i}^1\}_{i=1}^M \sim p(\theta) := p_\psi^0(\cdot|x_{\text{obs}})$. For all $r \geq 1$:

(i) Simulate new data $\{x_i^r\}_{i=1}^M \sim p(\cdot|\theta_{0,i}^r)$. Concatenate samples $\{(\theta_{0,i}^r, x_i^r)\}_{i=1}^M$ with those from previous rounds to form $\{(\theta_{0,i}, x_i)\}_{i=1}^{rM} := \bigcup_{s=1}^r \{(\theta_{0,i}^r, x_i^r)\}_{i=1}^M \sim \tilde{p}^r(\theta)p(x|\theta)$, where $\tilde{p}^r(\theta) = \frac{1}{r}\sum_{s=0}^{r-1} p_\psi^s(\theta|x_{\text{obs}})$. Draw times $\{t_i\}_{i=1}^{rM} \sim \mathcal{U}(0,T)$, and samples $\{\theta_{t_i,i}\}_{i=1}^{rM} \sim p_{t|0}(\cdot|\theta_{0,i})$.

(ii) Using these samples, train a time-varying score network $\tilde{s}_\psi^r(\theta_t, x, t)$ to approximate the score of the proposal posterior $\nabla_\theta \log \tilde{p}_t^r(\theta_t|x)$, by minimising a Monte Carlo estimate of

$$\mathcal{J}_{\text{post}}^{\text{DSM-C}}(\psi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)p(x|\theta_0)\tilde{p}^r(\theta_0)}\left[\left|\left|\tilde{s}_\psi^r(\theta_t, x, t) - \nabla_{\theta_t}\log p_{t|0}(\theta_t|\theta_0)\right|\right|^2\right]\mathrm{d}t, \tag{102}$$

where the score network $\tilde{s}_\psi^r(\theta_t, x, t)$ is defined as

$$\tilde{s}_\psi^r(\theta_t, x, t) = s_\psi(\theta_t, x, t) + \nabla_\theta \log \tilde{p}_t^r(\theta_t) - \nabla_\theta \log p_t(\theta_t). \tag{103}$$

or some approximation thereof (see Appendix C.4.3).

(iii) Draw samples $\{\theta_{T,i}^{r+1}\}_{i=1}^M \sim \pi(\theta)$. Simulate the backward SDE (3) or the probability flow ODE (4), substituting $s_\psi(\theta_t, x_{\text{obs}}, t)$ for $\nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$, to obtain samples $\{\theta_{0,i}^{r+1}\}_{i=1}^M \sim p_\psi^r(\cdot|x_{\text{obs}}) \approx p(\cdot|x_{\text{obs}})$.

Algorithm 5 SNPSE-C

```
Inputs: Observation $x_{\text{obs}}$, prior $p(\theta) =: p_\psi^0(\theta|x_{\text{obs}})$, simulator $p(x|\theta)$, simulation budget $N$, number of rounds $R$,
(simulations-per-round $M = N/R$)
Outputs: $p_\psi(\theta|x_{\text{obs}}) \approx p(\theta|x_{\text{obs}})$
Compute $\nabla_\theta \log p_t(\theta_t)$ or estimate $s_\psi^r(\theta_t, t) \approx \nabla_\theta \log p_t(\theta_t)$ (see Algorithm 2).
for $r = 1, \ldots, R$ do
  for $i = 1, \ldots, M$ do
    Draw $\theta_i \sim p_\psi^{r-1}(\theta|x_{\text{obs}})$, $x_i \sim p(x|\theta_i)$
    Add $(\theta_i, x_i)$ to $\mathcal{D}$
  end for
  if $r > 1$ then
    Learn $s_\varphi^{r,\text{prop}}(\theta_t, t) \approx \nabla_\theta \log \tilde{p}_t^r(\theta_t)$ by minimising (123).
    Define $\tilde{s}_\psi^r(\theta_t, x, t) := s_\psi(\theta_t, x, t) + s_\varphi^{r,\text{prop}}(\theta_t, t) - \nabla_\theta \log p_t(\theta_t)$
  else
    Define $\tilde{s}_\psi^r(\theta_t, x, t) := s_\psi(\theta_t, x, t)$
  end if
  Learn $\tilde{s}_\psi^r(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}_t^r(\theta_t|x)$ by minimising a Monte Carlo estimate of (102) based on dataset $\mathcal{D}$
  Get $p_\psi^r(\theta|x_{\text{obs}})$ sampler by substituting $s_\psi(\theta_t, x_{\text{obs}}, t) \approx \nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$ into (3) or (4).
end for
Return: $p_\psi^R(\theta|x_{\text{obs}})$
```

C.4.2. Theoretical Justification

We now outline the theoretical justification for SNPSE-C. We begin by noting that, via a repeated application of Bayes' Theorem, we have

$$p_t(\theta_t|x) = \int p_{t|0}(\theta_t|\theta_0)p(\theta_0|x)\mathrm{d}\theta_0 = \frac{p_t(\theta_t)\int_{\mathbb{R}^d} p_{0|t}(\theta_0|\theta_t)p(x|\theta_0)\mathrm{d}\theta_0}{p(x)} = \frac{p_t(\theta_t)p_t(x|\theta_t)}{p(x)}, \tag{104}$$

$$\tilde{p}^r_t(\theta_t|x) = \int p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0|x)\mathrm{d}\theta_0 = \frac{\tilde{p}^r_t(\theta_t)\int_{\mathbb{R}^d} \tilde{p}^r_{0|t}(\theta_0|\theta_t)p(x|\theta_0)\mathrm{d}\theta_0}{\tilde{p}(x)} = \frac{\tilde{p}^r_t(\theta_t)\tilde{p}^r_t(x|\theta_t)}{\tilde{p}(x)}, \tag{105}$$

where in the final equality in each line we have identified $p_t(x|\theta_t) = \int p(x|\theta_0)p_{0|t}(\theta_0|\theta_t)\mathrm{d}\theta_0$ and $\tilde{p}^r_t(x|\theta_t) = \int p(x|\theta_0)\tilde{p}^r_{0|t}(\theta_0|\theta_t)\mathrm{d}\theta_0$. It follows, taking logarithmic derivatives of (104) and (105), that

$$\nabla_\theta \log p_t(\theta_t|x) = \nabla_\theta \log p_t(x|\theta_t) + \nabla_\theta \log p_t(\theta_t), \tag{106}$$

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t|x) = \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) + \nabla_\theta \log \tilde{p}^r_t(\theta_t). \tag{107}$$

Thus, in particular, we can relate the perturbed posterior score $\nabla_\theta \log p_t(\theta_t|x)$ to the perturbed proposal posterior $\nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$ score according to

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t|x) = \nabla_\theta \log p_t(\theta_t|x) + \nabla_\theta \log \tilde{p}^r_t(\theta_t) - \nabla_\theta \log p_t(\theta_t) + \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) - \nabla_\theta \log p_t(x|\theta_t), \tag{108}$$

Now, suppose that we define $\tilde{s}^r_\psi(\theta_t, x, t)$ according to

$$\tilde{s}^r_\psi(\theta_t, x, t) = s_\psi(\theta_t, x, t) + \nabla_\theta \log \tilde{p}^r_t(\theta_t) - \nabla_\theta \log p_t(\theta_t) + \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) - \nabla_\theta \log p_t(x|\theta_t). \tag{109}$$

By standard results on conditional denoising score matching (e.g., Batzolis et al., 2021), we know that (102) is minimised by ψ^* such that $\tilde{s}^r_{\psi^*}(\theta, x, t) = \nabla_\theta \log \tilde{p}^r_t(\theta|x)$ for almost every $\theta \in \mathbb{R}^d$, $x \in \mathbb{R}^p$, and $t \in [0, T]$. It follows, using this observation, the definition in (109), and the identity in (108), that the minimiser $\psi^* = \mathrm{argmin}\mathcal{J}^{\mathrm{DSM-}C}_{\mathrm{post}}(\psi)$ of (102) is such that

$$s_{\psi^*}(\theta, x, t) = \tilde{s}^r_{\psi^*}(\theta_t, x, t) - \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log p_t(\theta_t) - \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) + \nabla_\theta \log p_t(x|\theta_t) \tag{110}$$

$$= \nabla_\theta \log \tilde{p}^r_t(\theta_t|x) - \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log p_t(\theta_t) - \nabla_\theta \log \tilde{p}^r_t(x|\theta_t) + \nabla_\theta \log p_t(x|\theta_t) \tag{111}$$

$$= \nabla_\theta \log p_t(\theta_t|x). \tag{112}$$

More generally, if we minimise (102) to obtain $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$, then we automatically recover $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$ via the definition (109). That is, we can automatically transform an estimate of the proposal posterior score into an estimate of the posterior score. This approach is reminiscent of SNPE-C (Greenberg et al., 2019), also referred to as automatic posterior transformation (APT).

In practice, the definition in (109) relies on several quantities - namely $\nabla_\theta \log \tilde{p}^r_t(\theta_t)$, $\nabla_\theta \log \tilde{p}^r_t(x|\theta_t)$ and $\nabla_\theta \log p_t(x|\theta_t)$ - to which we do not have immediate access. To make any progress, we will therefore need to make several simplifications and approximations. The first simplification is based on the observation that $\nabla_\theta \log p_0(x|\theta_0) = \nabla_\theta \log \tilde{p}_0(x|\theta_0)$ and $\nabla_\theta \log p_T(x|\theta_T) \approx \nabla_\theta \log \tilde{p}_T(x|\theta_T)$. Thus, substituting into (108), we have

$$\nabla_\theta \log \tilde{p}^r_0(\theta_0|x) = \nabla_\theta \log p_0(\theta_0|x) + \nabla_\theta \log \tilde{p}^r_0(\theta_0) - \nabla_\theta \log p_0(\theta_0), \tag{113}$$

$$\nabla_\theta \log \tilde{p}^r_T(\theta_T|x) \approx \nabla_\theta \log p_T(\theta_T|x) + \nabla_\theta \log \tilde{p}^r_T(\theta_T) - \nabla_\theta \log p_T(\theta_T). \tag{114}$$

Inspired by (113) - (114), suppose that we define a sequence of distributions $\{p^{r,\mathrm{seq}}_t(\theta_t|x)\}_{t\in[0,T]}$ according to $p^{r,\mathrm{seq}}_t(\theta_t|x) \propto \frac{p_t(\theta_t)}{\tilde{p}^r_t(\theta_t)}\tilde{p}^r_t(\theta_t|x) \propto \frac{\tilde{p}^r_t(x|\theta_t)}{p_t(x|\theta_t)}p_t(\theta_t|x)$. By construction, the scores of this sequence of distributions satisfy the identity

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t|x) = \nabla_\theta \log p^{r,\mathrm{seq}}_t(\theta_t|x) + \nabla_\theta \log \tilde{p}^r_t(\theta_t) - \nabla_\theta \log p_t(\theta_t). \tag{115}$$

In addition, based on (115), suppose that we now redefine the score network $\tilde{s}^r_\psi(\theta_t, x, t)$ according to

$$\tilde{s}^r_\psi(\theta_t, x, t) = s_\psi(\theta_t, x, t) + \nabla_\theta \log \tilde{p}^r_t(\theta_t) - \nabla_\theta \log p_t(\theta_t). \tag{116}$$

Then, arguing similarly to before, but now using (116) and (115) in place of (109) and (108), it follows that the minimiser $\psi^* = \text{argmin}\mathcal{J}_{\text{post}}^{\text{DSM}-C}(\psi)$ of (102) is such that

$$s_{\psi^*}(\theta, x, t) = \tilde{s}^r_{\psi^*}(\theta_t, x, t) - \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log p_t(\theta_t) \tag{117}$$

$$= \nabla_\theta \log \tilde{p}^r_t(\theta_t|x) - \nabla_\theta \log \tilde{p}^r_t(\theta_t) + \nabla_\theta \log p_t(\theta_t) = \nabla_\theta \log p_t^{r,\text{seq}}(\theta_t|x). \tag{118}$$

Thus, in general, if we minimise (102) to obtain $\tilde{s}^r_\psi(\theta_t, x, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t|x)$, then we automatically recover $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t^{r,\text{seq}}(\theta_t|x)$ via the definition in (116). Crucially, by our construction of $\{p_t^{r,\text{seq}}(\theta|x)\}_{t\in[0,T]}$, we have

$$p_0^{r,\text{seq}}(\theta_0|x) \propto \frac{p_0(\theta_0)}{\tilde{p}_0(\theta_0)}\tilde{p}^r_0(\theta_0|x) = \frac{p(\theta_0)}{\tilde{p}(\theta_0)}\tilde{p}(\theta_0|x) \propto p(\theta_0|x), \tag{119}$$

$$p_T^{r,\text{seq}}(\theta_T|x) \propto \frac{p_T(\theta_T)}{\tilde{p}_T(\theta_T)}\tilde{p}^r_T(\theta_T|x) \approx \frac{\mathcal{N}(\theta_T; 0, \mathbf{I})}{\mathcal{N}(\theta_T; 0, \mathbf{I})}\mathcal{N}(\theta_T; 0, \mathbf{I}) = \mathcal{N}(\theta_T; 0, \mathbf{I}). \tag{120}$$

Thus, in particular, $\{p_t^{r,\text{seq}}(\theta_t|x)\}_{t\in[0,T]}$ defines a sequence of distributions which smoothly interpolate between the true posterior $p_0(\theta|x) = p(\theta|x)$, and the reference distribution $\mathcal{N}(\theta; 0, \mathbf{I})$. In general, of course, $p_t^{r,\text{seq}}(\theta_t|x)$ will not coincide with $p_t(\theta_t|x)$, which means the sequence $\{p_t^{r,\text{seq}}(\theta_t|x)\}_{t\in[0,T]}$ does not correspond to the standard sequence $\{p_t(\theta_t|x)\}_{t\in[0,T]}$ of distributions obtained by applying the forward SDE (2), as described in Section 2.2. Thus, in particular, substituting the score network $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t^{r,\text{seq}}(\theta_t, x, t)$ obtained via (116) into the backward SDE (3) will not result in samples from the correct posterior distribution.

Nonetheless, based on our observation above that $\{p_t^{r,\text{seq}}(\theta_t|x)\}_{t\in[0,T]}$ smoothly interpolate between the true posterior $p_0(\theta|x) = p(\theta|x)$, and the reference distribution $p_T(\theta|x) = \mathcal{N}(\theta; 0, \mathbf{I})$, we can still use this score network to obtain samples from the correct posterior. In particular, by using $s_\psi(\theta_t, x, t) \approx \nabla_\theta \log p_t^{r,\text{seq}}(\theta_t, x, t)$ within an annealed MCMC algorithm, e.g., annealed Langevin dynamics (Song & Ermon, 2019; Geffner et al., 2023; Du et al., 2023), we should still recover samples from the correct posterior. See also Du et al. (2023, Section 4) for a related discussion, albeit in a very different context.

C.4.3. Estimating the Proposal Prior Score

Clearly, in order to define the score network $\tilde{s}^r_\psi(\theta_t, x, t)$ according to (103), we must be able to compute (or approximate) score of the perturbed proposal prior, namely,

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t) := \nabla_\theta \log \left[\frac{1}{r}\sum_{s=0}^{r-1} p^s_{\psi,t}(\theta_t|x_{\text{obs}})\right], \tag{121}$$

where $\tilde{p}^r_t(\theta_t) = \int_0^t p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0)\text{d}\theta_0$, $p^s_{\psi,t}(\theta_t|x_{\text{obs}}) = \int_0^t p_{t|0}(\theta_t|\theta_0)p^s_\psi(\theta_0|x_{\text{obs}})\text{d}\theta_0$; and $\tilde{p}^r(\theta)$ and $p^s_\psi(\theta|x_{\text{obs}})$ are defined as in Section C.4.1. Unfortunately, the score of the perturbed proposal prior cannot be written as a mixture of the score of the perturbed proposal priors from each of the previous rounds:

$$\nabla_\theta \log \tilde{p}^r_t(\theta_t) \neq \frac{1}{r}\sum_{s=0}^{r-1} \nabla_\theta \log p^s_{\psi,t}(\theta_t|x_{\text{obs}}) \tag{122}$$

That is, we cannot compute $\nabla_\theta \log \tilde{p}^r_t(\theta_t)$ in (121) using the score estimates obtained in previous rounds. Instead, to compute (121), and thus to use (103), we will need an alternative approach. Below, we outline several possibilities.

Approximating the Proposal Prior Score. The first and perhaps most natural approach is to approximate $s_\varphi^{r,\text{prop}}(\theta_t, t) \approx \nabla_\theta \log \tilde{p}^r_t(\theta_t)$ using a score network, and substitute this approximation into (103). Given samples $\theta \sim \tilde{p}^r(\theta)$, we can train this network by minimising a Monte Carlo estimate of the standard denoising score matching objective, viz,

$$\mathcal{J}_{\text{prop}}^{\text{DSM}-\text{C}}(\varphi) = \frac{1}{2}\int_0^T \lambda_t \mathbb{E}_{p_{t|0}(\theta_t|\theta_0)\tilde{p}^r(\theta_0)}\left[||s_\varphi^{r,\text{prop}}(\theta_t, t) - \nabla_\theta \log p_{t|0}(\theta_t|\theta_0)||^2\right] \text{d}t. \tag{123}$$

There are several advantages of this approach: it is conceptually very simple, and it fits rather naturally into our existing framework. At the same time, there are several disadvantages, particularly with regards to computational and memory costs.

First, to obtain a sufficiently accurate score network $s_{\varphi}^{r,\mathrm{prop}}(\theta_t,t)\approx\nabla_\theta\log\tilde{p}_t^r(\theta_t)$, it is desirable to use a large number of samples $\theta\sim\tilde{p}^r(\theta)$ from the proposal prior to form the Monte Carlo estimate of (123). By definition of the proposal prior, this requires generating (and storing) a large number of samples $\theta\sim p_\psi^s(\theta|x_{\mathrm{obs}})$, for $s=1,\dots,r-1$, which are obtained by repeated simulation of the relevant backward SDE (3) or probability flow ODE (4). Second, minimising (123) is itself a costly procedure, possibly requiring a large number of iterations in order to converge. The score network $s_{\varphi}^{r,\mathrm{prop}}(\theta_t,t)\approx\nabla_\theta\log\tilde{p}_t^r(\theta_t)$ must be re-learned after each round, which may amount to a significant additional computational cost over a large number of rounds.

Finally, the proposal prior score network $s_{\varphi}^{r,\mathrm{prop}}(\theta_t,t)\approx\nabla_\theta\log\tilde{p}_t^r(\theta_t)$ is substituted into (103) to compute $\tilde{s}_\psi^r(\theta_t,x,t)$, which is then learned by minimising the SNPSE-C objective in (102). Thus, minimising (102) requires an additional neural network pass at each training iteration. In addition, any error in the approximation $s_{\varphi}^{r,\mathrm{prop}}(\theta_t,t)\approx\nabla_\theta\log\tilde{p}_t^r(\theta_t)$ will adversely affect learning an accurate approximation $\tilde{s}_\psi^r(\theta_t,x,t)\approx\nabla_\theta\log\tilde{p}_t^r(\theta_t|x)$.

Approximating the Density Ratio Score. An alternative approach is instead to approximate the score of the ratio of the proposal prior density and the prior density, viz

$$\nabla_\theta\log\left[\frac{\tilde{p}_t^r(\theta_t)}{p_t(\theta_t)}\right]=\nabla_\theta\log\tilde{p}_t^r(\theta_t)-\nabla_\theta\log p_t(\theta_t). \tag{124}$$

Inspired by the recent perspective in Liu et al. (2024) on Stein variational gradient descent (Liu et al., 2016), we can approximate this term using a normalised, kernel-based estimate. In particular, let $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ be a positive semi-definite kernel. We can then approximate

$$\nabla_\theta\log\left[\frac{\tilde{p}_t^r(\theta_t)}{p_t(\theta_t)}\right]\approx\frac{\int_{\mathbb{R}^d}k(\theta_t,\theta_t')\nabla_{\theta'}\log\left[\frac{\tilde{p}_t^r(\theta_t')}{p_t(\theta_t')}\right]\tilde{p}_t^r(\theta_t')\mathrm{d}\theta_t'}{\int_{\mathbb{R}^d}k(\theta_t,\theta_t')\tilde{p}_t^r(\theta_t')\mathrm{d}\theta_t'} \tag{125}$$

$$=-\frac{\int_{\mathbb{R}^d}\left[\nabla_{\theta'}k(\theta_t,\theta_t')+\nabla_{\theta'}\log p_t(\theta_t')\right]\tilde{p}_t^r(\theta_t')\mathrm{d}\theta_t'}{\int_{\mathbb{R}^d}k(\theta_t,\theta_t')\tilde{p}_t^r(\theta_t')\mathrm{d}\theta_t'} \tag{126}$$

$$=-\frac{\mathbb{E}_{p_{t|0}(\theta_t'|\theta_0')\tilde{p}^r(\theta_0')}\left[\nabla_{\theta'}k(\theta_t,\theta_t')+\nabla_{\theta'}\log p_t(\theta_t')\right]}{\mathbb{E}_{p_{t|0}(\theta_t'|\theta_0')\tilde{p}^r(\theta_0')}\left[k(\theta_t,\theta_t')\right]}, \tag{127}$$

where the second line follows using integration by parts, and holds under mild regularity conditions (e.g., Liu et al., 2016; Korba et al., 2020). Crucially, the expectations in (127) only depend on samples $\theta_t'\sim p_{t|0}(\cdot|\theta_0')$, where $\theta_0'\sim\tilde{p}^r(\cdot)$, and can therefore be approximated using Monte Carlo.

Approximating the Proposal Prior. One final approach, somewhat different in spirit from the previous two, is to learn an approximation $\hat{p}^r(\theta)\approx\tilde{p}^r(\theta)$ of the proposal prior, or else an approximation $\hat{p}_\psi^s(\theta|x_{\mathrm{obs}})\approx p_\psi^s(\theta|x_{\mathrm{obs}})$ of each component of the proposal prior, for which it is possible to exactly compute the score of the corresponding perturbed proposal prior

$$\nabla_\theta\log\hat{p}_t^r(\theta_t):=\nabla_\theta\left[\frac{1}{r}\sum_{s=0}^{r-1}\hat{p}_{\psi,t}^s(\theta_t|x_{\mathrm{obs}})\right]. \tag{128}$$

There are several possible choices of surrogate proposal prior for which this calculation is possible. These include a mixture of Gaussians, a (continuous) uniform distribution, and an atomic (i.e., discrete uniform) distribution (e.g., Greenberg et al., 2019). We refer to Appendix B.2.1 for further details of how to compute the perturbed prior score in each of these cases. We leave further investigation of this approach to future work.

C.5. TSNPSE vs SNPSE-A vs SNPSE-B vs SNPSE-C

In Figure 6, we provide a comparison between TSNPSE, SNPSE-A, and SNPSE-B, for two of the benchmark tasks described in Lueckmann et al. (2021) (SLCP and GLU). We omit the corresponding results for SNPSE-C since, in our empirical testing, this method failed to provide meaningful results (e.g., C2ST ≈ 1). This, we suspect, is due to the significant approximation error incurred when estimating the score of the proposal prior, as described in Section C.4.3.

In both of these task, TSNPSE significantly outperforms both SNPSE-A and SNPSE-B, a finding which was also replicated in other tasks. We suspect that this is largely due to the error associated with the approximate importance weight correction used by SNPSE-A, and the high-variance updates associated with the use of importance weights in the loss function used by SNPSE-B. We note that the performance of SNPSE-B could likely be improved using the techniques recently introduced in Xiong et al. (2023).

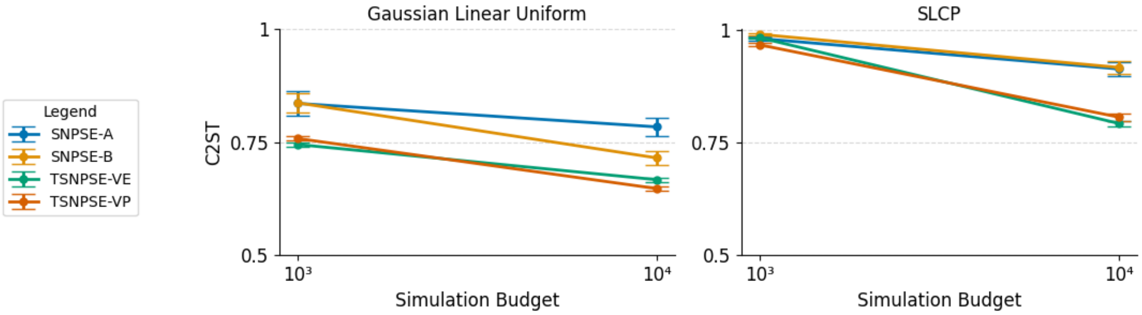


Figure 6. **Comparison between SNPSE-A, SNPSE-B, and TSNPSE on two benchmark tasks.**

D. Dealing with Multiple Observations

In this section, we discuss how to adapt our methods to the task of generating samples from $p(\theta_t|x^1_{\text{obs}},\ldots,x^n_{\text{obs}})$ for any set of observations $\{x^1_{\text{obs}},\ldots,x^n_{\text{obs}}\}$. As noted in Geffner et al. (2023), it is not possible to factorise the multiple-observation posterior score $\nabla_\theta \log p_t(\theta_t|x^1_{\text{obs}},\ldots,x^n_{\text{obs}})$ in terms of the single-observation posterior scores $\nabla_\theta \log p_t(\theta_t|x^i_{\text{obs}})$, and the prior score $\nabla_\theta \log p(\theta_t)$. Thus, a naive implementation of NPSE would require training a score network $s_\psi(\theta_t,x^1,\ldots,x^n,t)\approx\nabla_\theta \log p_t(\theta_t|x^1,\ldots,x^n)$ using samples $(\theta,x^1,\ldots,x^n)\sim p(\theta)\prod_{j=1}^n p(x^i|\theta)$. 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 $p(\theta|x^1,\ldots,x^n)\propto p(\theta)^{1-n}\prod_{i=1}^n p(\theta|x^i)$. In particular, they propose to use the sequence of densities

$$p_t^{(\text{bridge})}(\theta_t|x^1,\ldots,x^n)\propto(p(\theta_t)^{1-n})^{\frac{T-t}{T}}\prod_{i=1}^n p_t(\theta_t|x^i). \tag{129}$$

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

$$\nabla_\theta \log p_t^{(\text{bridge})}(\theta_t|x^1,\ldots,x^n)=\frac{(1-n)(T-t)}{T}\nabla_\theta \log p(\theta_t)+\sum_{i=1}^n\nabla_\theta \log p_t(\theta_t|x^i). \tag{130}$$

Thus, in particular, it is only necessary to learn a single score network $s_{\psi_{\text{post}}}(\theta_t,x,t)\approx\nabla_\theta \log p_t(\theta_t|x)$, 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

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

It is worth emphasising that, other than for $t=0$, $p_t^{(\text{bridge})}(\theta_t|x^1,\ldots,x^n)$ do not coincide with the true perturbed multi-observation posterior densities $p_t(\theta_t|x^1,\ldots,x^n)$. Thus, to generate samples, one must use an annealed MCMC algorithm (e.g., Geffner et al., 2023, Algorithm 1), rather than directly integrating the reverse-time SDE.

We now propose an alternative approach, based on a very similar idea to the one in Geffner et al. (2023). In particular, in place of (129), we now propose the sequence of densities

$$p_t^{(\text{bridge})}(\theta_t|x^1,\ldots,x^n)\propto(p_t(\theta_t))^{1-n}\prod_{i=1}^n p_t(\theta_t|x^i). \tag{131}$$

This sequence of densities has all of the desirable properties of (129). The density at $t=0$ coincides with the target $p(\theta_t|x^1,\ldots,x^n)$, and the density at $t=T$ is a tractable Gaussian. Moreover, we can factorise these densities in terms of the single-observation posterior scores $\nabla_\theta \log p_t(\theta_t|x^i)$, and the perturbed prior score $\nabla_\theta \log p_t(\theta_t)$, as

$$\nabla_\theta \log p_t^{(\text{bridge})}(\theta_t|x^1,\ldots,x^n)=(1-n)\nabla_\theta \log p_t(\theta_t)+\sum_{i=1}^n\nabla_\theta \log p_t(\theta_t|x^i). \tag{132}$$

Similar to above, it is then only necessary to learn a single score network $s_{\psi_{\text{post}}}(\theta_t, x, t) \approx \nabla_\theta \log p_t(\theta_t|x)$, which we can train using samples $(\theta, x) \sim p(\theta)p(x|\theta)$. Clearly, the expressions in (131) - (132) are very similar to the ones given in (129) - (130), with the only difference appearing in the first term. These quantities coincide at time zero, but will otherwise differ. The advantage of (130), i.e., the scheme proposed in Geffner et al. (2023), is that it only requires access to the score of the prior (rather than the score of the perturbed prior), and is thus very straightforward to implement.

E. Additional Experimental Details

E.1. Benchmark Tasks

We consider the following set of benchmark tasks, described in Lueckmann et al. (2021, Appendix T).

Gaussian Linear. This simple experiment involves inferring the mean of a 10-dimensional Gaussian, in which the covariance is fixed. The prior is a Gaussian, given by $p(\theta) = \mathcal{N}(0, 0.1\mathbf{I})$, as is the simulator, $p(x|\theta) = \mathcal{N}(x|\theta, 0.1\mathbf{I})$.

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 $p(x|\theta) = 0.5\mathcal{N}(x|\theta, \mathbf{I}) + 0.5\mathcal{N}(x|\theta, 0.01\mathbf{I})$, 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

$$x|\theta = \begin{pmatrix} r\cos(\alpha) + 0.25 \\ r\sin(\alpha) \end{pmatrix} + \begin{pmatrix} -|\theta_1 + \theta_2|/\sqrt{2} \\ (-\theta_1 + \theta_2)/\sqrt{2} \end{pmatrix} \tag{133}$$

where $\alpha \sim \mathcal{U}(-\pi/2, \pi/2)$ and $r \sim \mathcal{N}(0.1, 0.01^2)$. 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 methods deal with multimodality (Greenberg et al., 2019; Glockler et al., 2022).

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

Bernoulli GLM. This experiment consists of a generalised linear model (GLM) with Bernoulli observations, used to simulate the activity of a neuron depending on a single set of covariates (De Nicolao et al., 1997; Lueckmann et al., 2017). The task is to infer a 10-dimensional parameter $\theta = (\beta, \mathbf{f}) \in \mathbb{R}^{10}$, where $\beta \sim \mathcal{N}(0, 2)$ and $\mathbf{f} \sim \mathcal{N}(0, (\mathbf{F}^T\mathbf{F})^{-1})$, where $\mathbf{F}$ encourages smoothness by penalizing the second-order differences in the vector of parameters. The observed data $\mathbf{x} \in \mathbb{R}^{10}$ are the sufficient statistics for this GLM.

SLCP. This 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.

SIR. This is an epidemiological model in which individuals from a population move between 3 possible compartments: (S)usceptible, (I)nfected and (R)emoved. The task involves inferring a two-dimensional model parameter $\theta = (\beta, \gamma)$, where $\beta \sim \text{LogNormal}(\log(0.4), 0.5)$ is the contact rate between susceptible and infected, and $\gamma \sim \text{LogNormal}(\log(0.8), 0.2)$ is the mean removal rate. The data $x = (x_1, \ldots, x_{10}) \in \mathbb{R}^{10}$ consist of 10 equally spaced noisy recordings $x_i \sim \text{Bin}(1000, \frac{I}{N})$, where I denotes the number of individuals in the infected compartment, and is simulated according to a set of ODEs.

Lotka Volterra. This experiment involves a classical model used in ecology to model predator-prey populations (Lotka, 1920). The task is to infer a four dimensional parameter $\theta = (\alpha, \beta, \gamma, \delta)$ which governs the growth rates and the interactions of the predator and prey populations, which are described by a system of ODEs. The priors for these parameters are given by $\alpha, \gamma \sim \text{LogNormal}(-0.125, 0.5)$, and $\beta, \delta \sim \text{LogNormal}(-3, 0.5)$. The observed data $x = (x_1, \ldots, x_{10}) \in \mathbb{R}^{20}_+$, with each $x_i \in \mathbb{R}^2_+$, consist of 10 evenly spaced recordings of both predator and prey populations.

E.2. Real-World Neuroscience Problem

Additional Implementation Details. To deal with ill-defined summary statistics, we follow the approach adopted by Deistler et al. (2022a), and replace invalid summary statistics with a value two standard deviations below the prior predictive of the summary statistic. We use the VP SDE to diffuse samples (see Appendix E.3.1 for more details).

Additional Numerical Results. Additional results for this experiment are provided in Figures 7 - 8. In Figure 7, we provide a pairwise marginal plot for the posterior approximation obtained by TSNPSE. Our approximation has similar characteristics to those previously obtained in the literature; see, e.g., Deistler et al. (2022a) and Glockler et al. (2022). Meanwhile, Figure 8 shows the expected coverage of the approximate posterior, computed according to the simulation-based coverage calibration (SBCC) procedure described in Deistler et al. (2022a). This plot indicates that, for mid-low confidence levels, the empirical expected coverage is smaller than the confidence level (i.e., the posterior is overconfident). Importantly, however, the empirical expected coverage approximately matches the confidence level for high confidence levels. We expect that, as suggested in Hermans et al. (2022), an ensemble of approximate neural posteriors estimators could be used to obtain a more conservative posterior.

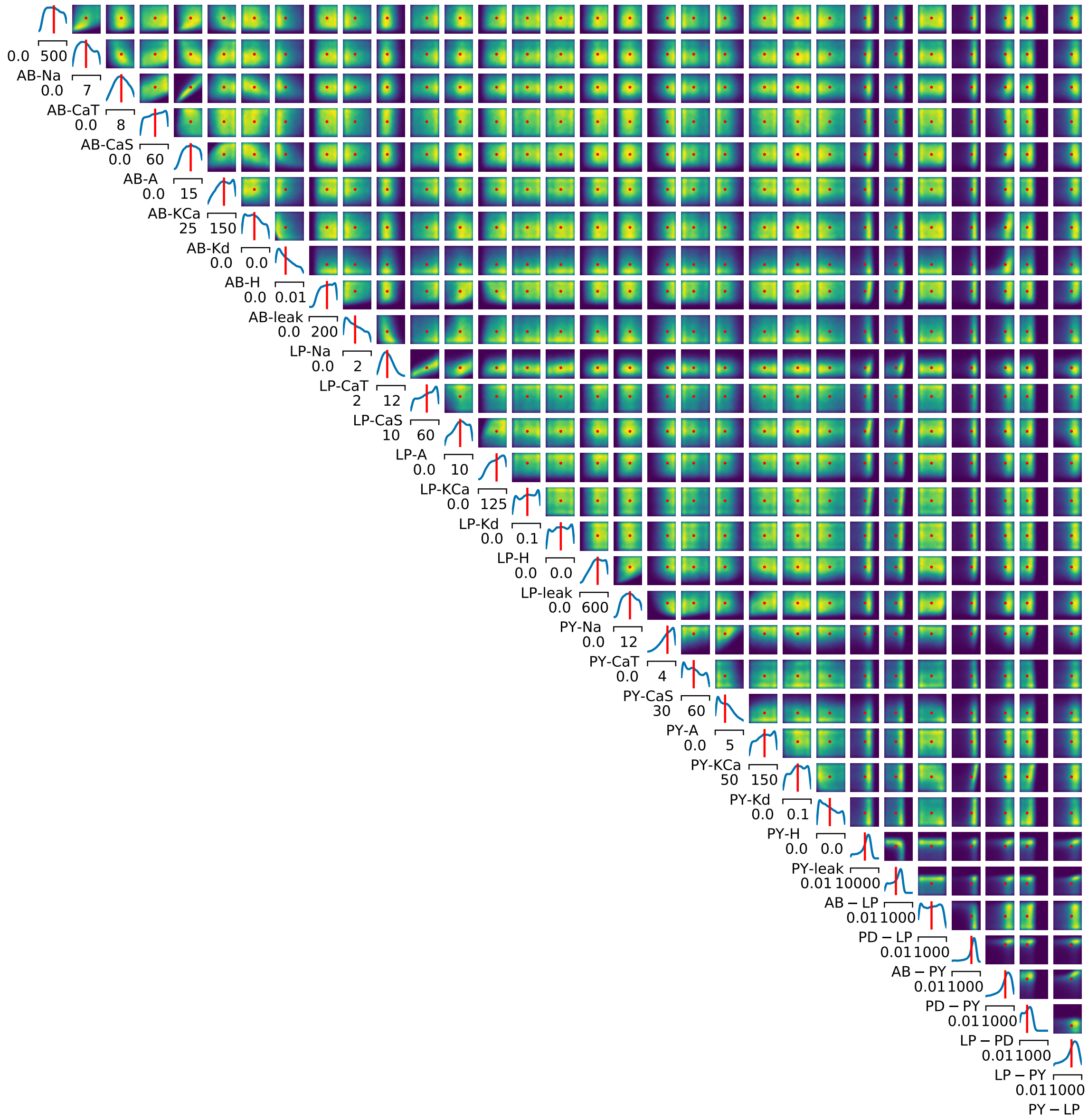


Figure 7. **Pairwise marginal plot for the posterior approximation obtained in the Pyloric experiment. The posterior mean is plotted in red.**

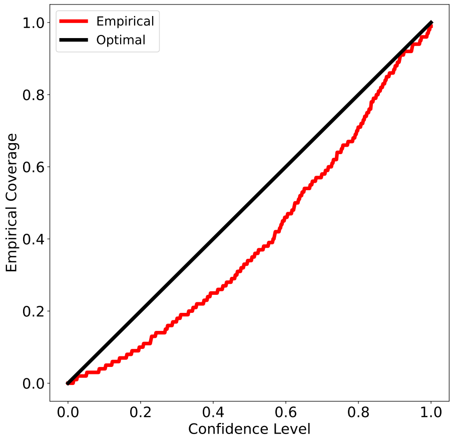


Figure 8. **Coverage plot for the Pyloric experiment.**

E.3. Implementation Details

E.3.1. SDE

For the benchmark experiments, we consider two different choices for the forward noising process: the variance exploding SDE (VE SDE) and the variance preserving SDE (VP SDE). See Song et al. (2021, Appendices B - C) for further details.

VE SDE. The VE SDE is defined according to

$$\mathrm{d}\theta_t = \sigma_{\text{min}}\left(\frac{\sigma_{\min}}{\sigma_{\max}}\right)^t\sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}\mathrm{d}w_t\ , \quad t\in(0,1]. \tag{134}$$

We set $\sigma_{\text{min}} = 0.01$ for the 2-dimensional experiments, SIR and Two Moons, and $\sigma_{\text{min}} = 0.05$ for all other experiments, while σ_{max} is chosen according to Technique 1 in Song & Ermon (2020). This SDE defines the transition density

$$p_{t|0}(\theta_t|\theta_0) = \mathcal{N}\left(\theta_t\,\middle|\,\theta_0, \sigma_{\text{min}}^2\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}\mathbf{I}\right). \tag{135}$$

VP SDE. The VP SDE is defined according to

$$\mathrm{d}\theta_t = -\frac{1}{2}\beta_t\theta_t\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}w_t\ , \quad t\in(0,1], \tag{136}$$

where $\beta_t = \beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})$. In our experiments, we set $\beta_{\text{min}} = 0.1$ and $\beta_{\text{max}} = 11.0$, following Song & Ermon (2020). This SDE defines the transition density

$$p_{t|0}(\theta_t|\theta_0) = \mathcal{N}\left(\theta_t\,\middle|\,\theta_0 e^{\frac{1}{2}\int_0^t\beta_s\mathrm{d}s}, \mathbf{I} - \mathbf{I}e^{\int_0^t\beta_s\mathrm{d}s}\right). \tag{137}$$

E.3.2. Network Architecture and Training

θ_t embedding network. 3-layer fully-connected MLP with 256 hidden units in each layer. The input dimension is d ($\theta\in\mathbb{R}^d$) and the output dimension from the final layer is determined by $\max(30, 4\cdot d)$. We denote this embedding θ_{emb}.

x embedding network. 3-layer MLP fully-connected with 256 hidden units in each layer. The input dimension is p ($x\in\mathbb{R}^p$) and the output dimension from the final layer is determined by $\max(30, 4\cdot p)$. We denote this embedding x_{emb}.

t sinusoidal embedding. We embed t into 64 dimensions, denoted t_{emb}. Inspired by Vaswani et al. (2017), we use sinusoidal embeddings defined by

$$(t_{\text{emb}})_i = \begin{cases} \sin\left(\frac{t}{10000^{(i-1)/31}}\right) & \text{if } i\leq 32,\\ \cos\left(\frac{t}{10000^{((i-32)-1)/31}}\right) & \text{if } i>32. \end{cases} \tag{138}$$

Score network. Finally, we concatenate $[\theta_{\text{emb}}, x_{\text{emb}}, t_{\text{emb}}]$ and feed this into a 3-layer fully-connected MLP with 256 hidden units in each layer whose output dimension is d.

Activation Function. We use SiLU activation functions between layers for all MLP networks.

Optimizer. We use Adam (Kingma & Ba, 2015) to train the networks with a learning rate of 1×10^{-4}. We hold 15% of the data back as a validation set: we compute the loss function on these samples after each training step, if this loss does not decrease for 1000 steps then we stop training and return the network which gave the lowest validation loss. The maximum number of training iterations is 3000 for sequential experiments and 3000 for non-sequential experiments. For experiments with a simulation budget of either 1000 or 10000, our batch size is 50 for non-sequential experiments and 200 for sequential experiments. For simulation budgets of 100000 we employ a bigger batch size of 500 for both sequential and non-sequential.

E.3.3. MISCELLANEOUS

Sampling. We use the probability flow ODE for sampling. To solve this ODE, we use an off-the-shelf solver (RK45).

Estimating HPR_ε and Sampling the Truncated Proposal. We follow the approach described in Deistler et al. (2022a, Section 6.3). We first simulate 20000 samples from our approximate posterior via (the time-reversal of) the probability flow ODE in (4) using our approximation of $\nabla_\theta \log p_t(\theta_t|x_{\text{obs}})$. We then compute the likelihood of these samples via the instantaneous change-of-variables formula in (5). Finally, we calculate the truncation boundary, κ, by taking the $(\varepsilon = 5 \times 10^{-4})^{\text{th}}$ quantile. This quantity defines the log-probability rejection threshold for rejection sampling.

To sample the truncated proposal, we use rejection sampling. In particular, we repeat the following procedure until the appropriate number of samples have been accepted: sample $\theta \sim p(\theta)$, compute the likelihood under our approximate posterior using the instantaneous change-of-variables formula in (5), and accept the sample if the likelihood is greater than κ, otherwise reject. As outlined in Deistler et al. (2022b, Section 3.2), one could also use other sampling schemes besides rejection sampling, such as Sampling Importance Resampling (SIR).

In practice, computing likelihoods via the instantaneous change of variables formula in (5) is a computationally expensive procedure, and thus we introduce an additional rejection sampling step to minimise the number of likelihood evaluations required.[4] In particular, we first perform a cheap initial rejection step directly on the prior samples by identifying whether they are within the (empirical) hypercube occupied by the approximate posterior samples. This is appropriate since the support of the approximate posterior typically takes up a fraction of the prior space; and can significantly reduce the number of likelihood calculations required.

Standardization. We centre both θ_t and x before being input into the score network, by subtracting an estimate of the mean and dividing by the standard deviation in each dimension.

Number of Rounds We use $R = 10$ rounds for all sequential experiments, unless otherwise specified. The simulation budget is equally divided between rounds.

F. Comparison to Flow Matching Posterior Estimation

In this section we provide an additional comparison between our non-sequential method (NPSE) and flow matching posterior estimation (FMPE) (Dax et al., 2023). In particular, results for the eight SBI benchmark experiments described in Section E.1, for simulation budgets of 1000, 10000, and 100000, are provided in Figure 9.

It is worth noting that the results for FMPE are taken directly from Dax et al. (2023). As a consequence, there is a difference in the way in which hyperparameters are tuned between the two methods. For FMPE, as detailed in Dax et al. (2023, Appendix C), hyperparameter tuning is performed on an experiment-by-experiment basis. This is achieved by sweeping over values of 5 hyperparameters (layer width, number of layers, learning rate, batch size, and a loss-weighting hyperparameter) for each experiment, and selecting the model which has the lowest validation loss on a held-out dataset. In contrast, NPSE uses a single set of hyperparameters for all experiments.

[4] See Appendix G for an alternative approach which reduces the cost of computing an (unnormalised) likelihood to the cost of a single forward pass of the neural network.

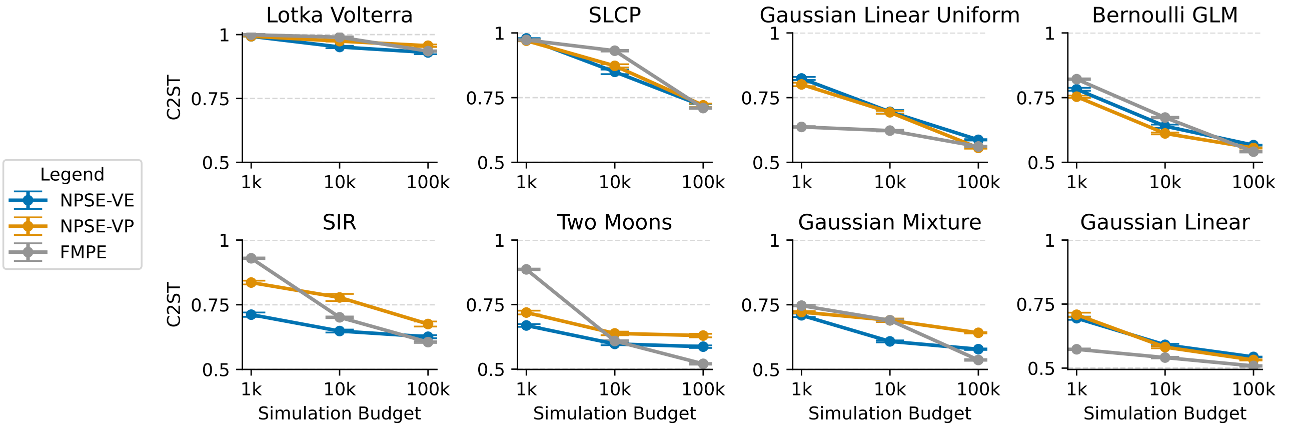


Figure 9. **Comparison between NPSE and FMPE on eight benchmark tasks.**

G. Alternative Parameterisation of the Score Network

In this section we discuss how an alternative energy-based parameterisation of the diffusion model (e.g., Du et al., 2023) can significantly reduce the computational cost of TSNPSE. Specifically, this parameterisation circumvents the need to use the instantaneous change-of-variables formula (5) to compute likelihoods, which is required for the truncated proposal used in TSNPSE (see Section E.3.3).

As discussed in Salimans & Ho (2021); Du et al. (2023), there are multiple ways to parameterise the posterior score estimate used in diffusion models. In this paper, we have directly modelled the score using a vector-valued score network $s_\psi : \mathbb{R}^d \times \mathbb{R}^p \times [0,T] \to \mathbb{R}^d$. As an alternative, one could parameterise a scalar-valued energy function $E_\psi : \mathbb{R}^d \times \mathbb{R}^p \times [0,T] \to \mathbb{R}$, and then model the score as the gradient of this function: $s_\psi(\theta_t, x, t) = -\nabla_{\theta_t} E_\psi(\theta_t, x, t)$.[5] In this case, we automatically also obtain an estimate of the perturbed posterior density as

$$p_t(\theta_t|x) \approx \frac{\exp\left[-E_\psi(\theta_t, x, t)\right]}{Z_\psi(x,t)}, \tag{139}$$

where $Z_\psi(x,t) = \int_{\mathbb{R}^d} \exp\left[-E_\psi(\theta_t, x, t)\right] \mathrm{d}\theta_t$ is an (unknown) normalising constant. Recalling that $p_0(\cdot|x) := p(\cdot|x)$, and noting that the normalising constant is independent of θ, it follows in particular that

$$p(\theta|x_{\text{obs}}) \overset{\propto}{\sim} \exp\left[-E_\psi(\theta, x_{\text{obs}}, 0)\right]. \tag{140}$$

Now, observe that estimating HPR_ε and sampling the truncated proposal (i.e., evaluating if samples are inside HPR_ε) actually only requires knowledge of the likelihood up to a normalisation constant (see Section E.3.3). Thus, under the energy-based parameterisation, estimating HPR_ε and sampling the truncated proposal only requires a single forward pass of the neural network E_ψ. This is significantly cheaper than using a score-based parameterisation, which not only requires multiple forward passes, but also the gradient of multiple forward passes, to solve (4) and (5).

It is worth noting that the choice of parameterisation can have a significant impact on the sample quality of (unconditional) score-based generative models; see, e.g., Salimans & Ho (2021, Section 4.1) or Du et al. (2023, Appendix E). It is likely that the same is true for the conditional score-based generative models used in TSNPSE. With this in mind, further empirical testing is required to understand whether the speed-up associated with the use of an energy-based parameterisation in TSNPSE comes at any cost in terms of sample quality.

[5] There are various ways in which one could parameterise the energy function E_ψ. The simplest is to parameterise E_ψ as a feedforward neural network, whose final layer has a single output (e.g., Nijkamp et al., 2020). Several other parameterisations have also been considered in Salimans & Ho (2021); Du et al. (2023).