Conditional Simulation Using Diffusion Schrödinger Bridges

Yuyang Shi\textsuperscript{1} Valentin De Bortoli\textsuperscript{2} George Deligiannidis\textsuperscript{1} Arnaud Doucet\textsuperscript{1}

\textsuperscript{1}Department of Statistics, University of Oxford, UK \textsuperscript{2}ENS, PSL University, Paris, France

\section*{Abstract}

Denoising diffusion models have recently emerged as a powerful class of generative models. They provide state-of-the-art results, not only for unconditional simulation, but also when used to solve conditional simulation problems arising in a wide range of inverse problems. A limitation of these models is that they are computationally intensive at generation time as they require simulating a diffusion process over a long time horizon. When performing unconditional simulation, a Schrödinger bridge formulation of generative modeling leads to a theoretically grounded algorithm shortening generation time which is complementary to other proposed acceleration techniques. We extend the Schrödinger bridge framework to conditional simulation. We demonstrate this novel methodology on various applications including image super-resolution, optimal filtering for state-space models and the refinement of pre-trained networks. Our code can be found at https://github.com/vdeborto/cdsb.

\section{1 INTRODUCTION}

Score-Based Generative Models (SGMs), also known as denoising diffusion models, are a class of generative models that have become recently very popular as they provide state-of-the-art performance; see e.g. Chen et al.\textsuperscript{2021a}, Ho et al.\textsuperscript{2020}, Song et al.\textsuperscript{2021b}, Saharia et al.\textsuperscript{2021}, Dhariwal and Nichol\textsuperscript{2021}. Existing SGMs proceed as follows. First, noise is gradually added to the data using a time-discretized diffusion so as to provide a sequence of perturbed data distributions eventually approximating an easy-to-sample reference distribution, typically a multivariate Gaussian. Second, one approximates the corresponding time-reversed denoising diffusion using neural network approximations of the logarithmic derivatives of the perturbed data distributions known as scores; these approximations are obtained using denoising score matching techniques [Vincent\textsuperscript{2011}, Hyvärinen\textsuperscript{2005}]. Finally, the generative model is obtained by initializing this reverse-time process using samples from the reference distribution [Ho et al.\textsuperscript{2020}, Song et al.\textsuperscript{2021b}].

In many applications, one is not interested in unconditional simulation but the generative model is used as an implicit prior $p_{\text{data}}(x)$ on some parameter $X$ (e.g. image) in a Bayesian inference problem with a likelihood function $g(y^{\text{obs}}|x)$ for observation $Y = y^{\text{obs}}$. SGMs have been extended to address such tasks, see e.g. Song et al.\textsuperscript{2021b}, Saharia et al.\textsuperscript{2021}, Batzolis et al.\textsuperscript{2021}, Tashiro et al.\textsuperscript{2021}. In this conditional simulation case, one only requires being able to simulate from the joint distribution of data and synthetic observations $(X, Y) \sim p_{\text{data}}(x)g(y|x)$. As in the unconditional case, the time-reversal of the noising diffusion is approximated using neural network estimates of its scores, the key difference being that this network admits not only $x$ but also $y$ as an input. Sampling from the posterior $p(x|y^{\text{obs}}) \propto p_{\text{data}}(x)g(y^{\text{obs}}|x)$ is achieved by simulating the time-reversal using the scores evaluated at $Y = y^{\text{obs}}$.

However, performing unconditional or conditional simulation using SGMs is computationally expensive as, to obtain a good approximation of the time-reversed diffusion, one needs to run the forward noising diffusion long enough to converge to the reference distribution. Many techniques have been proposed to accelerate simulation including e.g. knowledge distillation [Luhman and Luhman\textsuperscript{2021}, Salimans and Ho\textsuperscript{2022}], non-Markovian forward process and subsampling [Song et al.\textsuperscript{2021a}], optimized noising diffusions and improved numerical solvers [Jolicoeur-Martineau et al.\textsuperscript{2021}, Dockhorn et al.\textsuperscript{2022}, Kingma et al.\textsuperscript{2021}, Watson et al.\textsuperscript{2022}]. In the unconditional scenario, reformulating generative modeling as a Schrödinger bridge (SB) problem provides a principled theoretical framework to accelerate simulation time complementary to most other acceleration techniques [De Bortoli et al.\textsuperscript{2021}]. The SB solution is the
finite time process which is the closest in terms of Kullback–Leibler (KL) discrepancy to the forward noising process used by SGMs but admits as marginals the data distribution at time $t = 0$ and the reference distribution at time $t = T$. The time-reversal of the SB thus enables unconditional generation from the data distribution. However, the use of the SB formulation has not yet been developed in the context of conditional simulation.

The contributions of this paper are as follows.

- We develop conditional SB (CSB), an original SB formulation for conditional simulation.
- By adapting the Diffusion SB algorithm of De Bortoli et al. (2021) to our setting, we propose an iterative algorithm, Conditional Diffusion SB (CDSB), to approximate the solution to the CSB problem.
- CDSB performance is demonstrated on various examples. In particular, we propose the first application of score-based techniques to optimal filtering in state-space models.

## 2 SCORE-BASED GENERATIVE MODELING

### 2.1 UNCONDITIONAL SIMULATION

Assume we are given samples from some data distribution with positive density $p_{\text{data}}$ on $\mathbb{R}^d$. Our aim is to provide a generative model to sample new data from $p_{\text{data}}$. SGMs achieve this as follows. We gradually add noise to data samples, i.e. we consider a Markov chain $x_{0:N} = \{x_k\}_{k=0}^N \in \mathcal{X}$ of joint density

$$p(x_{0:N}) = p_0(x_0) \prod_{k=0}^{N-1} p_{k|k+1}(x_{k+1}|x_k),$$

where $p_0 = p_{\text{data}}$ and $p_{k|k+1}$ are Markov transition densities inducing the following marginal densities $p_{k+1}(x_{k+1}) = \int p_{k|k+1}(x_{k+1}|x_k)p_k(x_k)dx_k$. These transition densities are selected such that $p_N(x_N) \approx p_{\text{ref}}(x_N)$ for large $N$, where $p_{\text{ref}}$ is an easy-to-sample reference density. In practice we set $p_{\text{ref}}(x_N) = \mathcal{N}(x_N; 0, \text{Id})$, while $p_{k+1}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k - \gamma_{k+1} x_k; 2\gamma_{k+1} \text{Id})$ for $\gamma_{k+1} > 0$, $\gamma_k \ll 1$ so $x_{0:N}$ is a time-discretized Ornstein–Uhlenbeck diffusion (see supplementary details).

The main idea behind SGMs is to obtain samples from $p_0$ by exploiting the backward decomposition of $p_{\text{data}}$:

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_{k+1}|x_k),$$

i.e. by sampling $X_N \sim p_N(x_N)$ then sampling $X_k \sim p_{k|k+1}(x_{k+1}|X_{k+1})$ for $k \in \{N-1, \ldots, 0\}$, we obtain $X_0 \sim p_0(x_0)$. In practice, we know neither $p_N$ nor the backward transition densities $p_{k|k+1}$ for $k \in \{0, \ldots, N-1\}$ and therefore this ancestral sampling procedure cannot be implemented exactly. We thus approximate $p_N$ by $p_{\text{ref}}$ and $p_{k|k+1}$ using a Taylor expansion approximation

$$p_{k|k+1}(x|x_{k+1}) \approx \mathcal{N}(x; B_{k+1}(x_{k+1}), 2\gamma_{k+1} \text{Id}),$$

where $B_{k+1}(x) = x + \gamma_{k+1} \left[ x + 2\nabla \log p_{k|k+1}(x) \right]$. Finally, we approximate the score terms $\nabla \log p_k$ using denoising score matching methods (Hyvärinen, 2005; Vincent, 2011; Song et al., 2021b). Since $p_k(x_k) = \int p_0(x_0)p_{k|0}(x_k|x_0)dx_0$, it follows that $\nabla \log p_k(x_k) = \mathbb{E}[\nabla_x \log p_{k|0}(x_k|X_0)]$, where the expectation is w.r.t. to the distribution of $X_0$ given $x_k$. We learn a neural network approximation $s_{\theta}(k, x_k) \approx \nabla \log p_k(x_k)$ by minimizing w.r.t. $\theta$ the loss

$$\mathbb{E}\left[ \sum_{k=1}^{N} \lambda_k \|s_{\theta}(k, X_k) - \nabla x_k \log p_{k|0}(X_k|X_0)\|^2 \right],$$

where $\lambda_k > 0$ is a weighting coefficient (Ho et al., 2020; Song et al., 2021b) and the expectation is w.r.t. $p(x_0:N)$. Once we have estimated $\theta$ from noisy data, we start by first sampling $X_N \sim p_{\text{ref}}(x_N)$ and then sampling $X_k \sim p_{k|k+1}(x_{k+1}|X_{k+1})$ for $p_{k|k+1}$ as in $p_{k+1}$ but with $\nabla \log p_{k+1}(X_{k+1})$ replaced by $s_{\theta}(k+1, X_{k+1})$. Under regularity assumptions, the resulting $X_0$ can be shown to be approximately distributed according to $p_0 = p_{\text{data}}$ if $p_N \approx p_{\text{ref}}$ (De Bortoli et al., 2021, Theorem 1).

### 2.2 CONDITIONAL SIMULATION

We now consider the scenario where we have samples from $p_0 = p_{\text{data}}$ and are interested in generating samples from the posterior $p(x|y) \propto p_0(x)g(y|x)$ for some observation $Y = g(x) \in \mathcal{Y}$. Here it is assumed that it is possible to sample synthetic observations from $Y|X = x \sim g(y|x)$ but the expression of $g(y|x)$ might not be available.

In this case, conditional SGMs (CSGMs) proceed as follows; see e.g. Saharia et al. (2021), Batzolis et al. (2021), Li et al. (2022), Tashiro et al. (2021). For any realization $Y = y$, we consider a Markov chain of the form (1) but initialized using $X_0 \sim p(x|y)$ instead of $p_0(x)$. Obviously it is not possible to simulate this chain but this will not prove necessary. This chain induces for $k \geq 0$ the marginals denoted $p_{k+1}(x_{k+1}|y)$ which satisfy $p_{k+1}(x_{k+1}|y) = \int p_{k+1}(x_{k+1}|x_k)p_k(x_k|y)dx_k$ for $p_0(x_0|y) = p(x_0|y)$. Similarly to the unconditional case, to perform approximate ancestral sampling from this Markov chain, we need to sample from $p_{k+1}(x_{k+1}|x_k) \approx \mathcal{N}(x_{k+1}; B_{k+1}(x_{k+1}, y), 2\gamma_{k+1} \text{Id})$ where $B_{k+1}(x, y) = x + \gamma_{k+1} \left[ x + 2\nabla \log p_{k+1}(x|y) \right]$. We can again estimate these score terms using

$$\nabla \log p_k(x_k|y) = \mathbb{E}[\nabla_x \log p_k(x_k|X_0)],$$

where the expectation is w.r.t. to the distribution of $X_0$ given $(X_k, Y) = (x, y)$. In this case, we learn again a neural
network approximation $s_{θ'}(k, x_k, y) \approx \nabla \log p_k(x_k | y)$ by minimizing w.r.t. $θ$ the loss

$$\mathbb{E}[\sum_{k=1}^N \lambda_k \| s_{θ}(k, X_k, Y) - \nabla x_k \log p_k(0 | X_k, X_0) \|^2],$$

where the expectation is w.r.t. $p(x_0, N)g(y | x_0)$ which we can sample from. Once the neural network is trained, we simulate from the posterior $p(x | y^{\text{obs}}) \propto p_{θ}(x)g(y^{\text{obs}} | x)$ for any observation $Y = y^{\text{obs}}$ as follows: sample first $X_N \sim p_{\text{ref}}(X_N)$ and then $X_k \sim p_{θ(k+1)}(x_k | X_{k+1}, y^{\text{obs}})$ where this density is similar to $p_{θ(k+1)}(x_k | X_{k+1}, y^{\text{obs}})$ but with $\nabla \log p_k(1 | X_{k+1}, y^{\text{obs}})$ replaced by $s_{θ'}(k+1, X_{k+1}, y^{\text{obs}})$. The resulting sample $X_0$ will be approximately distributed according to $p(x | y^{\text{obs}})$. This scheme can be seen as an amortized variational inference procedure.

3 SCHRODINGER BRIDGES AND GENERATIVE MODELING

For SGMs to work well, we must diffuse the process long enough so that $p_N \approx p_{\text{ref}}$. The SB methodology introduced in [De Bortoli et al., 2021] allows us to mitigate this problem. We refer to [Chen et al., 2021b] for recent reviews on the SB problem. We first recall how the SB problem can be applied to perform unconditional simulation.

Consider the forward density $p(x_0, N)$ given by (1), describing the process adding noise to the data. We want to find the joint density $π^*(x_0, N)$ such that

$$π^* = \arg \min_π \{ \text{KL}(π | π^0) : p_0 = p_{\text{data}}, π_N = p_{\text{ref}} \}, \tag{2}$$

where $p_0$, resp. $π_N$, is the marginal of $X_0$, resp. $X_N$, under $π$. A visualization of the SB problem (2) is provided in Figure 1a. Were $π^*$ available, we would obtain a generative model by ancestral sampling: sample $X_N \sim p_{\text{ref}}(x_N)$, then $X_k \sim π_{θ(k+1)}(x_k | X_{k+1})$ for $k \in \{N - 1, \ldots, 0\}$.

The SB problem does not admit a closed-form solution but it can be solved numerically using Iterative Proportional Fitting (IPF) [Kullback, 1968]. This algorithm defines the following recursion initialized at $π^0 = p_0$ given in (1):

$$π_{2n+1} = \arg \min_π \{ \text{KL}(π | π_{2n}) : π_N = p_{\text{ref}} \}, π_{2n} = \arg \min_π \{ \text{KL}(π | π_{2n+1}) : π_0 = p_{\text{data}} \}.$$  

[De Bortoli et al., 2021], [Vargas et al., 2021] showed that the IPF iterates admit a representation suited to numerical approximation. Indeed, if we denote $π^n = π_{2n}$ and $q^n = π_{2n+1}$, then $p^n(x_0, N) = p^n(x_0, N)$ and

$$q^n(x_0, N) = p_{\text{ref}}(x_N) \prod_{k=0}^{N-1} q^n_{k+1} (x_k | x_{k+1}),$$

$$p^{n+1}(x_0, N) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} p_{θ(k+1)}^{n+1} (x_{k+1} | x_k),$$

where $q^n_{k+1} = p^n_{k+1} (x_{k+1})$ and $p^n_{k+1} (x_{k+1} | x_k)$. To summarize, at step $n = 0$, $q^0$ is the backward process obtained by reversing the dynamics of $p^0$ initialized at time $N$ from $p_{\text{ref}}$. The forward process $p^1$ is then obtained from the reversed dynamics of $q^0$ initialized at time 0 from $p_{\text{data}}$, and so on. Note that $q^n$ corresponds to the unconditional SGM described in Section 2.1.

3.1 DIFFUSION SCHRODINGER BRIDGE

Similarly to SGMs, one can approximate the time-reversals appearing in the IPF iterates using score matching ideas. If $p^n_{k+1}(x' | x) = N(x' ; x + γ_{k+1} f^n_k(x), 2γ_{k+1} I_d)$, with $f^n_k(x) = -x$, we approximate the reverse-time transitions by $q^n_{k+1}(x' | x) = N(x ; x' + γ_{k+1} b^n_{k+1}(x'), 2γ_{k+1} I_d)$, where $b^n_{k+1}(x') = -f^n_k(x') + 2\nabla \log p^n_{k+1}(x')$; and next $p^{n+1}_{k+1}(x' | x) = N(x' ; x + γ_{k+1} f^{n+1}_k(x'), 2γ_{k+1} I_d)$, where $f^{n+1}_k(x) = -b^n_{k+1}(x) + 2\nabla \log q^n_k(x)$. The drifts $b^n_{k+1}, f^{n+1}_k$ could be estimated by approximating $\{ \nabla \log p^n_{k+1}(x') \}_{i=0}^n, \{ \nabla \log q^n_k(x) \}_{i=0}^n$ using score matching. However this is too expensive both in terms of compute and memory. [De Bortoli et al., 2021] instead directly approximate the mean of the Gaussians using neural networks, $B_{θ}$ and $F_{θ}$, by generalizing the score matching approach, i.e. $q^n_{k+1}(x' | x) = N(x ; B_θ(x_k + 1, x'), 2γ_{k+1} I_d)$ and $p^{n+1}_{k+1}(x' | x) = N(x' ; F_θ(x, k, x), 2γ_{k+1} I_d)$, where $θ^n$ is obtained by minimizing

$$\ell^n(θ) = \mathbb{E}_{p_θ} [\sum_k \| B_θ(k + 1, X_{k+1}) - G_{n,k}(X_k, X_{k+1}) \|^2],$$

for $G_{n,k}(x, k, x') = x' + F_θ(x, k) - F_θ(x', k, x')$, and $ϕ^{n+1}$ by minimizing

$$\ell^{n+1}(ϕ) = \mathbb{E}_{p^n} [\sum_k \| F_θ(k, X_k) - H_{n,k}(X_k, X_{k+1}) \|^2],$$

for $H_{n,k}(x, k, x') = x + B_θ(k + 1, x', k) - B_θ(k + 1, x)$. This implementation of IPF, referred to as Diffusion SB (DSB), is presented in the supplementary; see [Vargas et al., 2021], [Chen et al., 2022] for alternative numerical schemes. After we have learned $θ^n$ using $L$ DSB iterations, we sample $X_N \sim p_{\text{ref}}(x_N)$ and then set $X_k = B_θ(k + 1, X_{k+1}) + \sqrt{2γ_{k+1}} Z_{k+1}$ with $Z_k \overset{i.i.d.}{\sim} N(0, I_d)$ to obtain $X_0$ approximately distributed from $p_{\text{data}}$.

3.2 LINK WITH OPTIMAL TRANSPORT

It can be shown that the solution $π^*$ of the SB problem (2), $π^*(x_0, x_N) = π^{n,*}(x_0, x_N)p_{θ,n}^{opt}(x_1, \ldots, x_{N-1} | x_0, x_N)$ where $π^{n,*}(x_0, x_N)$ is the marginal of $π^n(x_0, N)$ at times 0 and $N$. In this case, (2) reduces to the static SB problem

$$π^{n,*} = \arg \min_{π^n} \{ \text{KL}(π^n | π_0) : π^0_0 = p_0, π^N = p_{\text{ref}} \}.$$  

The static SB problem can be interpreted as an entropy-regularized optimal transport problem between $p_{\text{data}}$ and $p_{\text{ref}}$, with regularized transportation cost $\mathbb{E}_{π^n} [ -\log p_{N | 0}(X_N | X_0) ] - H(π^n)$. When $p_{N | 0}(x_N | x_0) = \delta(x_0)$, the cost becomes

$$\mathbb{E}_{π^n} [ -\log p_{N | 0}(X_N | X_0) ] = \lim_{\epsilon \to 0} \mathbb{E}_{π^n} [ \epsilon \log p_{N | 0}(X_N | X_0) ] = \lim_{\epsilon \to 0} -\mathbb{E}_{π^n} [ \epsilon \log p_{N | 0}(X_N | X_0) ] = \lim_{\epsilon \to 0} -\mathbb{E}_{π^n} [ \epsilon \log p_{N | 0}(X_N | X_0) ],$$

where $\epsilon$ is a small positive constant. This cost is minimized when $π^n = p_{\text{ref}}$ by the Kantorovich-Rubinstein duality [Kantorovich and Rubinstein, 1958].
We propose instead to solve an amortized problem. Let \( \pi \) where \( \frac{1}{2} \| x_0 - x \|^2 \) up to a constant. In other words, the static SB solution \( \pi^* \) not only transports samples \( X_N \sim \pi \) into samples from the data distribution \( p_{data} \), but also seeks to minimize an entropy-regularized Wasserstein distance of order 2. The regularization strength is controlled by the variance \( \sigma^2 \). Similar properties hold for the time-discretized Ornstein–Uhlenbeck diffusion defined by (1) in Section 2.1.

4 CONDITIONAL DIFFUSION SCHröDINGER BRIDGE

We now want to use SBs for conditional simulation, i.e. to be able sample from a posterior distribution \( p(x|y^{obs}) \propto p_{data}(x)g(y^{obs}|x) \) assuming only that it is possible to sample \( (X, Y) \sim p_{data}(x)g(y|x) \). In this case, an obvious approach would be to consider the SB problem where we replace \( p_{data}(x) \) by the posterior \( p(x|y^{obs}) \), i.e.

\[
\pi^* = \arg\min_\pi \left\{ KL(\pi | p_{p_{data}}) : \pi_0 = p(\cdot|y^{obs}), \pi_N = p_{\text{ref}} \right\}, \tag{3}
\]

where \( p_{p_{data}}(x_0) := p(x_0|y^{obs}) \frac{1}{\prod_{k=0}^{N-1} p_{k+1}(x_{k+1}|x_k)} \) is the forward noising process. However, DSB is not applicable here as it requires sampling from \( p(x_0|y^{obs}) \) at step 0.

We propose instead to solve an amortized problem. Let us introduce \( p_{p_{join}}(x, y) = p_{data}(x)g(y|x) = p(x|y)p_{\text{obs}}(y) \) and \( p_{p_{ref}}(x, y) = p_{\text{ref}}(x)p_{\text{obs}}(y) \) where \( p_{\text{obs}}(y) = \int p_{data}(x)g(y|x)dx \). We are interested in finding the transition kernel \( \pi^c, \pi^s \in \mathcal{Y} \), where \( \pi^c \) defines a distribution on \( \mathcal{X} = (\mathbb{R}^d)^{N+1} \) for each \( y \in \mathcal{Y} \), satisfying

\[
\pi^c = \arg\min_{\pi^c} \left\{ \mathbb{E}_{Y \sim p_{\text{obs}}}[KL(\pi^c | p_Y) | p_{\text{obs}}] : \pi_0^c \otimes p_{\text{obs}} = p_{p_{join}}, \pi_N^c \otimes p_{\text{obs}} = p_{\text{ref}} \right\}. \tag{4}
\]

This corresponds to an averaged version of (3) over the distribution \( p_{\text{obs}}(y) \) of \( Y \). The first constraint \( \pi^c_{\text{ref}}(x_0)p_{\text{obs}}(y) = p_{p_{join}}(x_0, y) = p(x_0|y)p_{\text{obs}}(y) \) ensures that \( \pi^c_{\text{ref}}(x_0) = p(x_0|y), \) \( p_{\text{obs}} \)-almost surely. Similarly \( \pi^s_{y,N}(x_N) = p_{\text{ref}}(x_N), \) \( p_{\text{obs}} \)-almost surely. Hence, to obtain a sample from \( p(x|y^{obs}) \) for a given \( Y = y^{obs} \), we can sample \( X_N \sim p_{\text{ref}}(x_N) \) then \( X_k|X_{k+1} \sim \pi^s_{y^{obs},k+1}(x_k|X_{k+1}) \) for \( k = N - 1, \ldots, 0 \) and \( X_0 \) is a sample from \( p(x|y^{obs}) \).

We show here that (3) can be reformulated as a SB on an extended space, which we will refer to as Conditional SB (CSB), so the theoretical results for existence and uniqueness of the solution to the SB problem apply.

**Proposition 1.** Consider the following SB problem

\[
\tilde{\pi}^* = \arg\min_{\tilde{\pi}} \left\{ KL(\tilde{\pi} | \tilde{\pi}^0) : \tilde{\pi}_0 = p_{p_{join}}, \tilde{\pi}_N = p_{p_{\text{ref}}} \right\}. \tag{5}
\]

where we define \( \tilde{p}(x_0,N, y_0,N) := p_{\text{ref}}(x_0, y_0) \pi_{\text{obs}}(y_0) \) with \( \pi_{\text{obs}}(y_0) := p_{\text{obs}}(y_0) \prod_{k=0}^{N-1} \delta_{y_k}(y_{k+1}) \) and \( y_0 \) is the forward process defined below (3). If \( KL(\tilde{\pi}^* | \bar{p}) < +\infty \) then \( \tilde{\pi}^* = \pi^c, \pi^s \) solves (4).

We provide an illustration of the CSB problem (5) in Figure 1b. Under \( \bar{p} \), the \( Y \)-component is sampled at time 0 according to \( p_{\text{obs}} \) and then is kept constant until time \( N \) while the \( X \)-component is initialized at \( p(x|y_0) \) and then differs according to \( p_{k+1}(x_{k+1}|x_k) \).

Contrary to (3), we can adapt DSB to solve numerically the CSB problem (5) as both the distributions \( p_{p_{join}} \) and \( p_{p_{ref}} \) can be sampled. The resulting algorithm is called Conditional DSB (CDSB). It approximates the following IPF recursion

\[
\bar{\pi}_{n+1} = \arg\min_{\tilde{\pi}} \left\{ KL(\tilde{\pi} | \bar{\pi}_{n+1}) : \tilde{\pi}_0 = \bar{\pi}_{n+1} \right\}.
\]

initialized at \( \tilde{\pi}_0 = \bar{p} \). For \( \bar{\pi}_{n} = \bar{\pi}_{n+1} \), we have the following representation of the IPF iterates.

**Proposition 2.** Assume that \( KL(p_{p_{join}} \otimes p_{p_{ref}} | p_{0,N}) < +\infty \). Then we have \( \bar{p}(x_0,N, y_0,N) = \bar{p}(x_0,N, y_0,N) \) and for any \( n > 0 \), \( \bar{\pi}_{n}(x_0,N, y_0,N) = p_{\text{obs}}(y_0)q^{n}(x_0,N|y_0,N) \), \( p_{n+1}(x_0,N, y_0,N) = p_{\text{obs}}(y_0)p_{n+1}(x_0,N|y_0) \) with

\[
q^n(x_0,N|y_N) = p_{\text{ref}}(x_N) \prod_{k=0}^{N-1} \delta_{x_{k+1}}(x_{k+1}|x_k, y_N),
\]

\[
p_{n+1}(x_0,N|y_0) = p(x_0|y_0) \prod_{k=0}^{N-1} q_{n+1}(x_{k+1}|x_{k+1}, y). \]
Here we simplify notation and write $Y$ for all the random variables $Y_0, Y_1, ..., Y_N$ as they are all equal almost surely under $\hat{p}^n$ and $\hat{q}^n$. We approximate the transition kernels as in DSB and refer to the supplementary for more details. In particular, the transition kernels satisfy

$$\hat{q}_{k+1}^n(x|x', y') = \mathcal{N}(x; \mathbf{B}_{\theta}^n(k + 1, x', 2\gamma_{k+1} \mathbf{I}_d)$$

and

$$\hat{p}_{k+1}^n(x|x, y) = \mathcal{N}(x'; \mathbf{F}_{\phi}^n(k, x), 2\gamma_{k+1} \mathbf{I}_d),$$

where $\theta^n$ is obtained by minimizing

$$\ell_n^q(\theta) = E_q^n(\sum_k \| \mathbf{B}_\theta^n(k + 1, X_{k+1}) - G_{n,k}(X_{k+1}) \|^2),$$

for $G_{n,k}(x, x') = x' + \mathbf{F}\phi^x_n(k, x) - \mathbf{F}\phi^x_n(k, x')$ and $\phi^{n+1}$ by minimizing

$$\ell_n^f(\phi) = E_q^n(\sum_k \| \mathbf{F}_\phi^n(k, x) - H_{n,k}(X_{k+1}, X_{k+1}) \|^2),$$

$$H_{n,k}(x, x') = x + \mathbf{B}_\phi^n(k + 1, x') - \mathbf{B}_\phi^n(k + 1, x),$$

5.1 CONDITIONAL REFERENCE MEASURE

In standard SGMs and for the unconditional SB, we typically select $p_{\text{ref}}(x) = \mathcal{N}(x; 0, \sigma_{\text{ref}}^2 \mathbf{I}_d)$. However, initializing ancestral sampling from random noise to eventually obtain samples from $p(x|y)$ can be inefficient as $y$ already contains useful information about $X$. Fortunately, it is easy to use a joint reference measure of the form $p_{\text{ref}}(x, y) = p_{\text{ref}}(x|y)p_{\text{obs}}(y)$ instead of $p_{\text{ref}}(x, y) = p_{\text{ref}}(x)p_{\text{obs}}(y)$ in CSB and CDSB. The only modification in Algorithm 1 is that line 8 becomes $Y \sim p_{\text{obs}}(y), X_N \sim p_{\text{ref}}(x|Y)$. In some interesting scenarios, we can select $p_{\text{ref}}(x|y)$ as an approximation to $p(x|y)$ in order to accelerate the sampling process. This means we construct a CSB between $p(x|y)$ and its approximation $p_{\text{ref}}(x|y)$, instead of between $p(x|y)$ and noise. We refer to this extension of CDSB as CDSB-C.

As a simple example, consider obtaining super-resolution (SR) image samples from a low-resolution image $Y = y$. Assume that $y$ has been suitably upsampled to have the same dimensionality as $X$. In this case, $y$ itself can serve as an approximate initialization for sampling $X_N$. A simple model is to take $p_{\text{ref}}(x|y) = \mathcal{N}(x; y, \sigma_0^2 \mathbf{I}_d)$ with $\sigma_0^2 = \rho \sigma_y^2$, where $\rho$ is a variance inflation parameter and $\sigma_y^2$ is an estimate of the conditional variance of $X$ given $Y$. See Figure 15 for an illustration. In our experiments, we also explore other $p_{\text{ref}}(x|y)$ obtained using the Ensemble Kalman Filter (EnKF) as well as neural network models.

5.2 CONDITIONAL FORWARD PROGRESS

To accelerate the convergence of IPF, we also have the flexibility to make the initial forward noising process dynamics dependent on $Y = y$, i.e. $p_y(x_{0:N}) = p(x_0|y)\prod_{k=0}^{N-1} p_{k+1}(k|x_{k+1}|x_k, y)$. As shown below, it is beneficial to initialize $p_y$ close to the CSB solution $\pi^*_y$.

**Proposition 3.** For any $n \in \mathbb{N}$ with $n \geq 1$, we have

$$E[KL(\pi^{c,n}_y(p(\cdot|Y)))] \leq \frac{1}{n} E[KL(\pi^*_y(p_Y))],$$

where for any $n \in \mathbb{N}$, $\pi^n = p_{\text{obs}} \circ \pi^{c,n}$ is the $n^{th}$ IPF iterate and the expectations are w.r.t. $Y \sim p_{\text{obs}}$.

As a result, we should choose the initial forward noising process $p_y$ such that its terminal marginal $p_y, N$ targets $p_{\text{ref}}(\cdot|y)$. However, contrary to diffusion models, we recall that our framework does not strictly require $p_y, N \approx p_{\text{ref}}(\cdot|y)$ to provide approximate samples from the posterior of interest.

For tractable $p_{\text{ref}}(x|y)$, we can define $p_y(x_{0:N})$ using an unadjusted Langevin dynamics; i.e. $p_{k+1}(k|x', y) = \mathcal{N}(x'; x + \gamma_{k+1} \nabla \log p_{\text{ref}}(x|y), 2\gamma_{k+1} \mathbf{I}_d)$. In the case $p_{\text{ref}}(x|y) = \mathcal{N}(x; \mu(y), \sigma^2(y) \mathbf{I}_d)$, this reduces to a discretized Ornstein–Uhlenbeck process admitting $p_{\text{ref}}(x|y)$ as limiting distribution as $\gamma \to 0$ and $N \to \infty$ [Durmus and Moulines, 2017].

5.3 FORWARD-BACKWARD SAMPLING

When we use an unconditional $p_{\text{ref}}(x)$, our proposed method also shares connections with the conditional transport

1796
methodology developed by [Marzouk et al. 2016, Spantini et al. 2022]. They propose methods to learn a deterministic invertible transport map \( S(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{X} \) which maps samples from \( p(x|y) \) to \( p_{\text{ref}}(x) \). To sample from \( p(x|y_{\text{obs}}) \), one samples \( X_{\text{ref}} \sim p_{\text{ref}}(x) \), then transports back the sample through the inverse map \( X_{\text{pos}} = S(\cdot, y_{\text{obs}})^{-1}(X_{\text{ref}}) \).

As noted by [Spantini et al. 2022], an alternative method to sample from \( p(x|y_{\text{obs}}) \) consists of first sampling \( (X, Y) \sim p_{\text{poin}} \), then following the two-step transformation \( X_{\text{ref}} = S(X, Y) \), \( X_{\text{pos}} = S(\cdot, y_{\text{obs}})^{-1}(X_{\text{ref}}) \). By definition of \( S \), \( X_{\text{ref}} \) is also distributed according to \( p_{\text{ref}} \). However, since the transport map \( S \) may be imperfect in practice, this sampling strategy provides the advantage of cancellation of errors between \( S \) and \( S(\cdot, y_{\text{obs}})^{-1} \).

We also explore an analogous forward-backward sampling scheme in our framework, which first samples \( (X, Y) \sim p_{\text{poin}} \), followed by sampling \( X \sim \tilde{p}_{\text{N}}(x|X, Y) \) through the forward half-bridge, then \( \tilde{X}_0 \sim \tilde{q}^L_{\text{G}}(x_0|\tilde{X}_N, y_{\text{obs}}) \) through the backward half-bridge. Since \( \tilde{q}^L \) is the approximate time-reversal of \( p^L \), this strategy shares similar advantages as the method of [Spantini et al. 2022] when the half-bridge \( \tilde{q}^L(x_0, N|y_{\text{obs}}) \) does not solve the CSB problem exactly. We call this extension CDSB-FB.

6 RELATED WORK

Approximate Bayesian computation (ABC), also known as likelihood-free inference, has been developed to approximate the posterior when the likelihood is intractable but one can simulate synthetic data from it; see e.g. [Beaumont 2019]. However, these methods typically require knowing the prior, while CDSB only needs to have access to joint samples and learns about the posterior directly. For tasks such as image inpainting, the prior is indeed implicit.

Schrödinger bridges techniques to perform both static and sequential Bayesian inference for state-space models have been developed by [Bernton et al. 2019] and [Reich 2019]. However, these methods require being able to evaluate pointwise an unnormalized version of the target posterior distribution contrary to the CDSB-based methods developed here.

Conditional transport. Performing conditional simulation by learning a transport map between joint distributions on \( X, Y \) having the same \( Y \)-marginals (as \( p_{\text{poin}} \) and \( p_{\text{ref}} \)) has been first proposed by [Marzouk et al. 2016]. Various techniques have been subsequently developed to approximate such maps such as polynomial or radial basis representations [Marzouk et al. 2016, Baptista et al. 2020]. Generative Adversarial Networks [Kovachki et al. 2021, Zhou et al. 2022] or normalizing flows [Kruse et al. 2021]. CDSB also fits into this framework, but instead utilizes stochastic transport maps. Recently, [Laghaei and Hosseini 2022] have also proposed independently using conditional transport ideas to perform optimal filtering for state-space models.

Conditional SGMs. SGMs have been applied to perform posterior simulation, primarily for images, as described in Section 2.2 and references therein. An alternative line of work for image editing [Song and Ermon 2019, Choi et al. 2021, Chung et al. 2021, Meng et al. 2022] utilizes the denoising property of SGMs to iteratively denoise noisy versions of a reference image \( y \) while restricted to retain particular features of \( y \). However, \( p_{\text{ref}}(x) = \mathcal{N}(x; 0, \sigma^2_{\text{ref}} I_d) \) so image generation is started from noise and typically hundreds or thousands of refinement steps are required. Our framework can incorporate in a principled way information given by \( y \) in the reverse process’s initialization (see Section 5.1). Recently [Zheng et al. 2022, Lu et al. 2022] have also proposed suitable choices for \( p_{\text{ref}}(x) \) or \( p_{\text{ref}}(x|y) \) to shorten the diffusion process. In comparison, the CDSB framework is more flexible and allows for general \( p_{\text{ref}}(x|y) \) which can be non-Gaussian and different from the initial forward diffusion’s terminal distribution \( p_{\text{ref}}(x_0) \). For instance, we explore using noiseless pre-trained super-resolution models as \( p_{\text{ref}}(x|y) \) in Section 7.3.2 where CDSB further improves the SR samples closer to the data distribution. Finally, for linear Gaussian inverse problems, [Kadkhodaee and Simoncelli 2021, Kawar et al. 2021, 2022] develop efficient methodologies using unconditional SGMs when the linear degradation model and the Gaussian noise level are known.

SGM acceleration techniques. Many techniques have been proposed to accelerate SGMs and CSGMs. For example, [Luhman and Luhman 2021, Salimans and Ho 2022] propose to learn a distillation network on top of SGM models, while [Song et al. 2021a] perform a subsampling of the timesteps in a variational setting. [Watson et al. 2022] optimize the timesteps with a fixed budget using dynamic programming. [Xiao et al. 2021] perform multi-steps denoising using GANs while [Dockhorn et al. 2022] consider underdamped Langevin dynamics as forward process. We emphasize that many of these techniques are complementary to and can be readily applied in the SB setting; e.g. one could distill the last CDSB network \( B_{\theta_k} \). Additionally, SB and CSB provide a framework to perform few-step sampling.

7 EXPERIMENTS

7.1 2D SYNTHETIC EXAMPLES

We first demonstrate the validity and accuracy of our method using the two-dimensional examples of [Kovachki et al. 2021]. We consider three nonlinear, non-Gaussian examples for \( p_{\text{poin}}(x, y) \): define \( p_{\text{obs}}(y) = \text{Unif}(y; [-3, 3]) \) for all examples and \( p(x|y) \) is defined through

Example 1: \( X = \tanh(Y) + Z \), \( Z \sim \Gamma(1, 0.3) \).

Example 2: \( X = \tanh(Y + Z) \), \( Z \sim \mathcal{N}(0, 0.05) \).

Example 3: \( X = Z \tanh(Y) \), \( Z \sim \Gamma(1, 0.3) \).
We now consider a Bayesian inference problem on biochemical oxygen demand (BOD) from Marzouk et al. [2016].

Table 1: Estimated posterior moments for the BOD example. The closest estimates to MCMC are highlighted in bold.

<table>
<thead>
<tr>
<th></th>
<th>MCMC</th>
<th>CDSB</th>
<th>CDSB-FB</th>
<th>CDSB-C</th>
<th>MGAN</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>.075</td>
<td>.066</td>
<td>.068</td>
<td>.072</td>
<td>.048</td>
<td>.034</td>
</tr>
<tr>
<td>$x_2$</td>
<td>.875</td>
<td>.897</td>
<td>.897</td>
<td>.891</td>
<td>.918</td>
<td>.902</td>
</tr>
<tr>
<td>Var</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>.190</td>
<td>.184</td>
<td>.190</td>
<td>.188</td>
<td>.177</td>
<td>.206</td>
</tr>
<tr>
<td>$x_2$</td>
<td>.397</td>
<td>.387</td>
<td>.391</td>
<td>.393</td>
<td>.419</td>
<td>.457</td>
</tr>
<tr>
<td>Skew</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>1.94</td>
<td>1.90</td>
<td>2.01</td>
<td>1.90</td>
<td>1.83</td>
<td>1.63</td>
</tr>
<tr>
<td>$x_2$</td>
<td>.681</td>
<td>.591</td>
<td>.628</td>
<td>.596</td>
<td>.630</td>
<td>.872</td>
</tr>
<tr>
<td>Kurt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>8.54</td>
<td>7.85</td>
<td>8.54</td>
<td>8.00</td>
<td>7.64</td>
<td>7.57</td>
</tr>
<tr>
<td>$x_2$</td>
<td>3.44</td>
<td>3.33</td>
<td>3.51</td>
<td>3.27</td>
<td>3.19</td>
<td>3.88</td>
</tr>
</tbody>
</table>

Figure 2: True posterior $p(x|y^{\text{obs}})$ for $y^{\text{obs}} \in \{-1.2, 0, 1.2\}$ (solid lines) and approximations for the 2D examples.

Table 1 displays moment statistics of the estimated posterior $p(x|y)$ (standard deviations are reported in the supplementary), in comparison with the “ground truth” statistics computed using $6 \times 10^6$ MCMC steps as reported in Marzouk et al. [2016]. To match the evaluation in Kovachki et al. [2021], the reported statistics are computed using 30,000 samples and averaged across the last 10 CDSB iterations. The resulting posterior displays high skewness and high kurtosis, but all CDSB-based methods achieve more accurate posterior estimation than MGAN and the inverse transport (IT) method in Marzouk et al. [2016].

7.3 IMAGE EXPERIMENTS

7.3.1 Gaussian Reference Measure

We now apply CDSB to a range of inverse problems on image datasets. We consider the following tasks: (a) MNIST 4x SR (7x7 to 28x28), (b) MNIST center 14x14 inpainting, (c) CelebA 4x SR (16x16 to 64x64) with Gaussian noise of $\sigma_y = 0.1$, (d) CelebA center 32x32 inpainting. For CSGM-C and CDSB-C, we consider the following choices for conditional $p_{\text{ref}}(x|y)$: for tasks (a) and (c), we use the upsampled $y$ directly as described in Section 5.1; for inpainting tasks (b) and (d), we use a separate neural network with the same architecture as $F, B$ to output the initialization mean. In Table 2 we report PSNR and SSIM (the higher the better), as well as FID scores (the lower the better) for RGB images only. We display a visual comparison between the methods in Figures 3 and 4 and additional image samples in the supplementary. CDSB and CDSB-C both provide significant improvement in terms of quantitative metrics as well as visual evaluations, and high-quality images can be generated quickly under few iterations $N$.

7.3.2 Pre-trained SR Model for Reference Measure

We further explore here the possibility of using a non-Gaussian $p_{\text{ref}}(x|y)$ to further bridge the gap towards the true posterior $p(x|y)$. We utilize the super-resolution model SRFlow [Lugmayr et al. 2020], which produces a probability distribution over possible SR images using a conditional normalizing flow. We use their pre-trained model checkpoints for the 8x SR task for CelebA (160x160). We then train a short CDSB model with SRFlow as $p_{\text{ref}}(x|y)$, in order to take advantage of the high sampling quality of diffusion models. As can be seen from Figure 5 with only $N = 10$ steps the CDSB model is able to make meaningful improvements to the SRFlow samples, especially in the finer details such as facial features and hair texture. Quantitatively, CDSB-C produces significant improvement over
Table 2: Results for (a) MNIST 4x SR; (b) MNIST 14x14 inpainting; (c) CelebA 4x SR with Gaussian noise; (d) CelebA 32x32 inpainting. Reported results are denoted in the format PSNR↑/SSIM↑/(FID↓).

<table>
<thead>
<tr>
<th></th>
<th>$N = 5$</th>
<th>$N = 10$</th>
<th>$N = 10$</th>
<th>$N = 20$</th>
<th>$N = 20$</th>
<th>$N = 50$</th>
<th>$N = 20$</th>
<th>$N = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSGM</td>
<td>17.22/0.672</td>
<td>20.03/0.795</td>
<td>14.77/0.599</td>
<td>16.31/0.706</td>
<td>19.52/0.471</td>
<td>20.02/0.567</td>
<td>20.52/0.567</td>
<td>24.82/0.844</td>
</tr>
<tr>
<td>CDSB</td>
<td>18.55/0.746</td>
<td>20.69/0.792</td>
<td>16.24/0.618</td>
<td>16.61/0.657</td>
<td>19.72/0.504</td>
<td>20.70/0.590</td>
<td>20.70/0.590</td>
<td>24.88/0.850</td>
</tr>
<tr>
<td>CSGM-C</td>
<td>18.61/0.749</td>
<td>20.83/0.838</td>
<td>16.38/0.701</td>
<td>16.53/0.730</td>
<td>20.44/0.566</td>
<td>20.84/0.592</td>
<td>22.89</td>
<td></td>
</tr>
<tr>
<td>CDSB-C</td>
<td>19.67/0.753</td>
<td>20.95/0.840</td>
<td>16.60/0.700</td>
<td>16.65/0.747</td>
<td>21.11/0.614</td>
<td>21.46/0.646</td>
<td>13.71</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Uncurated samples for the MNIST 4x SR task with $N = 5$.

Figure 4: Uncurated samples for the CelebA 4x SR with Gaussian noise task with $N = 20$.

7.4 FILTERING IN STATE-SPACE MODELS

Consider a state-space model defined by a bivariate Markov chain $(X_t, Y_t)_{t \geq 1}$ of initial density $\mu(x_1)g(y_1|x_1)$ and transition density $f(x_{t+1}|x_t)g(y_{t+1}|x_{t+1})$ where $X_t$ is latent while $Y_t$ is observed. We are interested in estimating sequentially in time the filtering distribution $p(x_t|y_{1:t}^{\text{obs}})$, that is the posterior of $X_t$ given the observations $Y_{1:t} = y_{1:t}^{\text{obs}}$. We show here how CDSB can be used at each time $t$ to obtain a sample approximation of these filtering distributions. This CDSB-based algorithm only requires us being able to sample from the transition density $f(x_{t+1}|x_t)g(y_{t+1}|x_{t+1})$ and is thus more generally applicable than standard techniques such as particle filters [Doucet and Johansen, 2009].

Assume at time $t$, one has a collection of samples $\{X_{i,t}^{1:M}\}_{i=1}^{M}$ distributed (approximately) according to $p(x_t|y_{1:t}^{\text{obs}})$. We sample $X_{t+1}^{i} \sim f(x_{t+1}|X_{t}^{i})$ and $Y_{t+1}^{i} \sim g(y_{t+1}|X_{t+1}^{i})$. The resulting samples $\{X_{t+1}^{i}, Y_{t+1}^{i}\}_{i=1}^{M}$ are thus distributed according to $p_{\text{pot}}(x_{t+1}, y_{t+1}) := p(x_{t+1}, y_{t+1}|y_{1:t}^{\text{obs}})$. We can also easily obtain samples from $p_{\text{pot}}(x_{t+1}, y_{t+1}) := p_{\text{pot}}(x_{t+1}|y_{t+1}, y_{1:t}^{\text{obs}})p(y_{t+1}|y_{1:t}^{\text{obs}})$, where $p_{\text{pot}}(x_{t+1}|y_{t+1}, y_{1:t}^{\text{obs}})$ is an easy-to-sample distribution designed by the user. Thus we can use CDSB to obtain a (stochastic) transport map between $p_{\text{pot}}(x_{t+1}, y_{t+1})$ and $p_{\text{pot}}(x_{t+1}, y_{t+1})$ and applying it to $Y_{t+1} = y_{t+1}^{\text{obs}}$, we can obtain new samples from $p(x_{t+1}|y_{1:t+1}^{\text{obs}})$. A similar stra-
CDSB-C achieves the lowest error consistently. When using $N = 100$, CSGM can achieve RMSE comparable with CDSB-C using $N = 20$, but CDSB still provides advantages compared to CSGM. CSGM-C achieves comparable RMSE as CDSB-C with suitably long diffusion process in this case. For lower ensemble size, e.g. $M = 200$, occasional large errors occur for some of the runs; see supplementary for details. We conjecture that this is due to overfitting.

Table 4: RMSEs over 10 runs between each algorithm’s filtering means and the ground truth filtering means for $N = 20$ (short) and $N = 100$ (long).

<table>
<thead>
<tr>
<th>$M$</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnKF</td>
<td>.354±.006</td>
<td>.355±.005</td>
<td>.354±.003</td>
</tr>
<tr>
<td>CSGM(C) (short)</td>
<td>Diverges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDSB (short)</td>
<td>.251±.011</td>
<td>.218±.008</td>
<td>.196±.005</td>
</tr>
<tr>
<td>CDSB-C (short)</td>
<td>.236±.012</td>
<td>.207±.014</td>
<td>.178±.007</td>
</tr>
<tr>
<td>CSGM (long)</td>
<td>.232±.008</td>
<td>.203±.009</td>
<td>.182±.009</td>
</tr>
<tr>
<td>CDSB (long)</td>
<td>.220±.012</td>
<td>.195±.007</td>
<td>.166±.004</td>
</tr>
<tr>
<td>CSGM-C (long)</td>
<td>.210±.009</td>
<td>.185±.005</td>
<td>.162±.004</td>
</tr>
<tr>
<td>CDSB-C (long)</td>
<td>.218±.014</td>
<td>.185±.008</td>
<td>.160±.003</td>
</tr>
</tbody>
</table>

8 DISCUSSION

We have proposed a SB formulation of conditional simulation and an algorithm, CDSB, to approximate its solution. The first iteration of CDSB coincides with CSGM while subsequent ones can be thought of as refining it. This theoretically grounded approach is complementary to the many other techniques that have been recently proposed to accelerate SGMs and could be used in conjunction with them. However, it also suffers from limitations. As CDSB approximates numerically the diffusion processes output by IPF, the minimum $N$ one can pick to obtain reliable approximations is related to the steepness of the drift of these iterates which is practically unknown. Additionally CSGM and CDSB are only using $y_{\text{obs}}$ when we want to sample from $p(x|y_{\text{obs}})$ but not at the training stage. Hence if $y_{\text{obs}}$ is not an observation “typical” under $p_{\text{obs}}(y)$, the approximation of the posterior can be unreliable. In the ABC context, the best available methods rely on procedures which sample synthetic observations in the neighbourhood of $y_{\text{obs}}$. It would be interesting but challenging to extend such ideas to CSGM and CDSB. Other interesting potential extensions include developing an amortized version of CDSB for filtering that would avoid having to solve a SB problem at each time step, and a conditional version of the multimarginal SB problem.

Acknowledgements

We thank James Thornton for his helpful comments. We are also grateful to the authors of [Kovachki et al., 2021] for sharing their code with us.
References


Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In International Conference on Learning Representations, 2021b.


