Conditional Simulation Using Diffusion Schrödinger Bridges

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 such as image inpainting or deblurring. 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 here the Schrödinger bridge framework to conditional simulation. We demonstrate this novel methodology on various applications including image super-resolution and optimal filtering for state-space models.

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. [2021a], Ho et al. [2020], Song et al. [2021b], Saharia et al. [2021], Dhariwal and Nichol [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, 2011]. Finally, the generative model is obtained by initializing this reverse-time process using samples from the reference distribution [Ho et al., 2020, Song et al., 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. [2021b], Saharia et al. [2021], Batzolis et al. [2021], Tashiro et al. [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, 2021, Salimans and Ho, 2022], subsampling [Song et al., 2021a], optimized noising diffusions and improved numerical solvers [Dockhorn et al., 2022, Jolicoeur-Martineau et al., 2021, Kingma et al., 2021, Watson et al., 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., 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

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

reference distribution at time $t = T$. The time-reversal of the SB thus enables unconditional generation from the data distribution using shorter time intervals than standard SGMs. 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.
- We show how it is possible to exploit existing approximate inference techniques to speed up sample generation both theoretically and empirically.
- 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[1] p_{data} on $\mathbb{R}^d$. Our aim is to provide a generative model to sample new data from p_{data}. SGMs achieve this as follows. We gradually add noise to data samples, i.e. we consider a Markov chain $x_{0:N} = \{x_k\}_{k=0}^N \in \mathcal{X} = (\mathbb{R}^d)^{N+1}$ of joint density

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

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

The main idea behind SGMs is to obtain samples from p_0 by exploiting the backward decomposition of (1)

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}), \qquad (2)$$

i.e. by sampling $X_N \sim p_N(x_N)$ then sampling $X_k \sim p_{k|k+1}(x_k|X_{k+1})$ 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 $k \in \{0, ..., N-1\}$ and therefore this ancestral sampling procedure cannot be implemented exactly. We thus approximate p_N by p_{ref} and $p_{k|k+1}$ using a Taylor expansion approximation

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

where $B_{k+1}(x, \gamma) = x + \gamma\{x + 2\nabla \log p_{k+1}(x)\}$. Finally, we approximate the score terms $\nabla \log p_k$ using denoising score matching methods [Hyvärinen and Dayan, 2005, Vincent, 2011, Song et al., 2021b]. Since $p_k(x_k) = \int p_0(x_0) p_{k|0}(x_k|x_0) \mathrm{d}x_0$, it follows that

$$\nabla \log p_k(x_k) = \mathbb{E}[\nabla_{x_k} \log p_{k|0}(x_k|X_0)],$$

where the expectation is w.r.t. to the distribution of X_0 given x_k. We learn a neural network approximation $\mathbf{s}_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k)$ by minimizing w.r.t. θ the loss

$$\mathbb{E}[\textstyle\sum_{k=1}^N \lambda_k ||\mathbf{s}_\theta(k, X_k) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)||^2],$$

where $\lambda_k > 0$ is a weighting coefficient Ho et al. [2020], Song et al. [2021b] and the expectation is w.r.t. to $p(x_{0:N})$. Once we have estimated $\theta^\star$, $\mathbf{s}_{\theta^\star}$ from noisy data, our generative model mimicks (2) by first sampling $X_N \sim p_{\text{ref}}(x_N)$ and then sampling $X_k \sim \hat{p}_{k|k+1}(x_k|X_{k+1})$ for $\hat{p}_{k|k+1}$ as in $p_{k|k+1}$ but with $\nabla \log p_{k+1}(X_{k+1})$ replaced by $\mathbf{s}_{\theta^\star}(k+1, X_{k+1})$. 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 $p(x|y^{\text{obs}}) \propto p_0(x) g(y^{\text{obs}}|x)$ for some observation $Y = y^{\text{obs}} \in \mathcal{Y}$. 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], 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 $p_{k+1}(x_{k+1}|y) = \int p_{k+1|k}(x_{k+1}|x_k) p_k(x_k|y) \mathrm{d}x_k$. Similarly to the unconditional case, to perform approximate ancestral sampling from this Markov chain, we need to sample from $p_{k|k+1}(x_k|x_{k+1}, y) \approx \mathcal{N}(x_k; B_{k+1}(x_{k+1}, \gamma_{k+1}, y), 2\gamma_{k+1}\,\mathrm{Id})$ where we have $B_{k+1}(x, \gamma, y) = x + \gamma\{x + 2\nabla \log p_{k+1}(x|y)\}$. We can again estimate these score terms using

$$\nabla \log p_k(x_k|y) = \mathbb{E}[\nabla_{x_k} \log p_{k|0}(x_k|X_0)],$$

[1] We assume here that all distributions admit a positive density w.r.t. Lebesgue measure.

where the expectation is w.r.t. to the distribution of X_0 given $(X_k, Y) = (x_k, y)$. In this case, we learn again a neural network approximation $\mathbf{s}_{\theta^\star}(k, x_k, y) \approx \nabla \log p_k(x_k|y)$ by minimizing w.r.t. θ the loss

$$\mathbb{E}[\sum_{k=1}^N \lambda_k ||\mathbf{s}_\theta(k, X_k, Y) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)||^2],$$

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 can simulate from the posterior $p(x|y^{\text{obs}}) \propto p_0(x)g(y^{\text{obs}}|x)$ for any observation $Y = y^{\text{obs}}$ as follows: sample first $X_N \sim p_{\text{ref}}(x_N)$ and then $X_k \sim \hat{p}_{k|k+1}(x_k|X_{k+1}, y^{\text{obs}})$ where this density is similar to $p_{k|k+1}(x_k|X_{k+1}, y^{\text{obs}})$ but with $\nabla \log p_{k+1}(X_{k+1}|y^{\text{obs}})$ replaced by $\mathbf{s}_{\theta^\star}(k+1, X_{k+1}, y^{\text{obs}})$. The resulting sample X_0 will be appproximately distributed according to $p(x|y^{\text{obs}})$. This scheme can be thought of as an amortized variational inference procedure; see e.g. Batzolis et al. [2021].

3 SCHRÖDINGER BRIDGES AND GENERATIVE MODELING

For SGMs to work well, we must diffuse the process long enough for $p_N \approx p_{\text{ref}}$. The SB methodology introduced in [De Bortoli et al., 2021] allows us to mitigate this problem. We refer to Léonard [2014b], 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. Then we detail our novel contribution: the extension of the SB framework to conditional simulation.

3.1 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 $\pi^\star(x_{0:N})$ such that

$$\pi^\star = \arg\min_\pi \left\{\text{KL}(\pi|p) \,:\, \pi_0 = p_{\text{data}},\ \pi_N = p_{\text{ref}}\right\},$$

where π_0, resp. π_N, is the marginal of X_0, resp. X_N, under π. Were $\pi^\star$ available, we would obtain a generative model by ancestral sampling: sample $X_N \sim p_{\text{ref}}(x_N)$, then $X_k \sim \pi^\star_{k|k+1}(x_k|X_{k+1})$ for $k \in \{N-1, \dots, 0\}$. 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 $\pi^0 = p$ given in (1):

$$\pi^{2n+1} = \arg\min_\pi \left\{\text{KL}(\pi|\pi^{2n}) \,:\, \pi_N = p_{\text{ref}}\right\},$$
$$\pi^{2n+2} = \arg\min_\pi \left\{\text{KL}(\pi|\pi^{2n+1}) \,:\, \pi_0 = p_{\text{data}}\right\}.$$

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 $p^n = \pi^{2n}$ and $q^n = \pi^{2n+1}$, then $p^0(x_{0:N}) = p(x_{0:N})$ and

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

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

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

Similarly to SGMs, one can approximate the time-reversals using score matching ideas. If $p^n_{k+1|k}(x'|x) = \mathcal{N}(x'; x + \gamma_{k+1} f^n_k(x), 2\gamma_{k+1}\,\text{Id})$, with $p^0 = p$ and $f^0_k(x) = -x$, we approximate the reverse-time transitions by $q^n_{k|k+1}(x|x') = p^n_{k|k+1}(x|x') \approx \mathcal{N}(x; x' + \gamma_{k+1} b^n_{k+1}(x'), 2\gamma_{k+1}\,\text{Id})$, where $b^n_{k+1}(x') = -f^n_k(x') + 2\nabla \log p^n_{k+1}(x')$, and the forward transitions with $p^{n+1}_{k+1|k}(x'|x) = q^n_{k+1|k}(x'|x) \approx \mathcal{N}(x'; x + \gamma_{k+1} f^{n+1}_k(x), 2\gamma_{k+1}\,\text{Id})$, where $f^{n+1}_k(x) = -b^n_{k+1}(x) + 2\nabla \log q^n_k(x)$. The drifts f^{n+1}_k, b^{n+1}_k could be estimated by approximating $\{\nabla \log p^i_{k+1}(x)\}_{i=0}^n$, $\{\nabla \log q^i_k(x)\}_{i=0}^n$ 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, $\mathbf{F}_\phi$ and $\mathbf{B}_\theta$, by generalizing the score matching approach, i.e. $q^n_{k|k+1}(x|x') = \mathcal{N}(x; \mathbf{B}_{\theta^n}(k+1, x'), 2\gamma_{k+1}\,\text{Id})$ and $p^n_{k+1|k}(x'|x) = \mathcal{N}(x'; \mathbf{F}_{\phi^n}(k, x), 2\gamma_{k+1}\,\text{Id})$ where θ^n is obtained by minimizing

$$\ell^b_n(\theta) = \mathbb{E}_{p^n}[\textstyle\sum_k \|\mathbf{B}_\theta(k+1, X_{k+1}) - G_{n,k}(X_k, X_{k+1})\|^2], \tag{3}$$

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

$$\ell^f_{n+1}(\phi) = \mathbb{E}_{q^n}[\textstyle\sum_k \|\mathbf{F}_\phi(k, X_k) - H_{n,k}(X_k, X_{k+1})\|^2], \tag{4}$$

for $H_{n,k}(x, x') = x + \mathbf{B}_{\theta^n}(k+1, x') - \mathbf{B}_{\theta^n}(k+1, x)$. This implementation of IPF, refered 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^L using L DSB steps, we sample $X_N \sim p_{\text{ref}}(x_N)$ and then set $X_k = \mathbf{B}_{\theta^L}(k+1, X_{k+1}) + \sqrt{2\gamma_{k+1}} Z_{k+1}$ with $Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$ to obtain X_0 approximately distributed from p_{data}.

3.2 CONDITIONAL SIMULATION

We now want to use SBs for conditional simulation, i.e. to be able sample from a posterior distribution $p(x|y^{\text{obs}}) \propto p_{\text{data}}(x)g(y^{\text{obs}}|x)$ assuming only that it is possible to sample

$(X, Y) \sim p_{\text{data}}(x)g(y|x)$. In this case, an obvious approach would be to consider the SB problem where we replace $p_{\text{data}}(x)$ by the posterior $p(x|y^{\text{obs}})$, i.e.

$$\pi^\star = \arg\min_\pi \left\{ \text{KL}(\pi|p_{y^{\text{obs}}}) : \pi_0 = p(\cdot|y^{\text{obs}}),\ \pi_N = p_{\text{ref}} \right\}, \tag{5}$$

with $p_{y^{\text{obs}}}(x_{0:n}) := p(x|y^{\text{obs}}) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k)$. However, DSB is not applicable here as it requires sampling from $p_{y^{\text{obs}}}(x_{0:n})$ and thus $p(x_0|y^{\text{obs}})$ at step 0.

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

$$\begin{aligned}\pi^{c,\star} = \text{argmin}_{\pi^c} \{ &\mathbb{E}_{Y\sim p_{\text{obs}}}[\text{KL}(\pi^c_Y||p_Y)] : \\ &\pi^c_0 \otimes p_{\text{obs}} = p_{\text{join}},\ \pi^c_N \otimes p_{\text{obs}} = p_{\text{jref}} \}.\end{aligned} \tag{6}$$

This corresponds to solving an averaged version of (5) over the unconditional distribution p_{obs} of Y. As the solution to this problem satisfies the constraint $\pi^{c,\star}_{y,0}(x_0)p_{\text{obs}}(y) = p(x_0|y)p_{\text{obs}}(y)$, this means that $\pi^{c,\star}_{y,0}(x_0) = p(x_0|y)$ p_{obs}-almost surely. We also have that $\pi^{c,\star}_{y,N}(x_N) = p_{\text{ref}}(x_N)$ p_{obs}-almost surely. Hence, to obtain a sample from $p(x|y^{\text{obs}})$ for a given $Y = y^{\text{obs}}$, we can sample $X_N \sim \pi^{c,\star}_{y^{\text{obs}},N}(x_N)$ then $X_k|X_{k+1} \sim \pi^{c,\star}_{y^{\text{obs}},k|k+1}(x_k|X_{k+1})$ for $k = N-1, ..., 0$ and X_0 is a sample from $p(x|y^{\text{obs}})$.

We show here that (6) 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. Additionally, we can adapt DSB to solve numerically the CSB problem. The resulting algorithm is called Conditional DSB (CDSB).

Proposition 1. *Consider the following SB problem*

$$\bar\pi^\star = \text{argmin}_{\bar\pi}\{\text{KL}(\bar\pi|\bar p) : \textit{s.t. } \bar\pi_0 = p_{\text{join}},\ \bar\pi_N = p_{\text{jref}}\}, \tag{7}$$

where we define $\bar p(x_{0:N}, y_{0:N}) := p_{y_0}(x_{0:N})\bar p_{\text{obs}}(y_{0:N})$ *and* $\bar p_{\text{obs}}(y_{0:N}) := p_{\text{obs}}(y_0)\prod_{k=0}^{N-1}\delta_{y_k}(y_{k+1})$. *Assume that* $\text{KL}(\bar\pi^\star|\bar p) < +\infty$. *Then, we have* $\bar\pi^\star = \pi^{c,\star} \otimes \bar p_{\text{obs}}$ *where* $\pi^{c,\star}$ *solves* (6).

Under $\bar p$, the Y-component is sampled at time 0 according to p_{obs} and then is kept constant until time N while the X-component is initialized at $p(x|y_0)$ and then diffuses according to $p_{k+1|k}$ to p_{ref}. Contrary to (5), we can adapt DSB to solve (7) as both the distributions p_{join} and p_{jref} can be sampled. In more detail, the IPF procedure solving (7) and hence (6) is given by the recursion initialized at $\bar\pi^0 = \bar p$ and for $n \geq 0$

$$\begin{aligned}\bar\pi^{2n+1} &= \arg\min_{\bar\pi}\left\{\text{KL}(\bar\pi|\bar\pi^{2n}) \ :\ \bar\pi_N = p_{\text{jref}}\right\}, \\ \bar\pi^{2n+2} &= \arg\min_{\bar\pi}\left\{\text{KL}(\bar\pi|\bar\pi^{2n+1}) \ :\ \bar\pi_0 = p_{\text{join}}\right\}.\end{aligned}$$

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

$$\begin{aligned}\bar q^n(x_{0:N}|y_N) &= p_{\text{ref}}(x_N)\textstyle\prod_{k=0}^{N-1}\bar p^n_{k|k+1}(x_k|x_{k+1}, y_N), \\ \bar p^{n+1}(x_{0:N}|y_0) &= p(x_0|y_N)\textstyle\prod_{k=0}^{N-1}\bar q^n_{k+1|k}(x_{k+1}|x_k, y_0).\end{aligned}$$

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 $\bar p^n$ and $\bar q^n$. We approximate the transition kernels as in DSB and refer to the supplementary for more details. In particular, the transition kernels satisfy $\bar q^n_{k|k+1}(x|x', y) = \mathcal{N}(x; \mathbf{B}^y_{\theta^n}(k+1, x'), 2\gamma_{k+1}\,\text{Id})$ and $\bar p^n_{k+1|k}(x'|x, y) = \mathcal{N}(x'; \mathbf{F}^y_{\phi^n}(k, x), 2\gamma_{k+1}\,\text{Id})$ where θ^n is obtained by minimizing

$$\ell^b_n(\theta) = \mathbb{E}_{\bar p^n}[\textstyle\sum_k \|\mathbf{B}^Y_\theta(k+1, X_{k+1}) - G^Y_{n,k}(X_k, X_{k+1})\|^2] \tag{8}$$

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

$$\begin{aligned}\ell^f_{n+1}(\phi) &= \mathbb{E}_{\bar q^n}[\textstyle\sum_k \|\mathbf{F}^Y_\phi(k, X_k) - H^Y_{n,k}(X_k, X_{k+1})\|^2], \\ H^y_{n,k}(x, x') &= x + \mathbf{B}^y_{\theta^n}(k+1, x') - \mathbf{B}^y_{\theta^n}(k+1, x).\end{aligned} \tag{9}$$

The resulting CDSB scheme is summarized in Algorithm 1 where $Z^j_k, \tilde Z^j_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$. After L iterations of CDSB, we have learned θ^L. For any observation $Y = y^{\text{obs}}$, we can then sample approximately from the posterior $p(x|y^{\text{obs}})$ by sampling $X_N \sim p_{\text{ref}}(x_N)$ and then use $X_k = \mathbf{B}^{y^{\text{obs}}}_{\theta^L}(k+1, X_{k+1}) + \sqrt{2\gamma_{k+1}}Z_{k+1}$ with $Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$ for $k = N-1, ..., 0$. The resulting sample X_0 will be approximately distributed from $p(x|y^{\text{obs}})$.

Algorithm 1 Conditional Diffusion Schrödinger Bridge

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample $\{X^j_k\}^{N,M}_{k,j=0}, \{Y^j\}^M_{j=0}$ where $X^j_0 \sim p_{\text{data}}, Y^j \sim g(\cdot|X^j_0)$, and $X^j_{k+1} = \mathbf{F}^{Y^j}_{\phi^n}(k, X^j_k) + \sqrt{2\gamma_{k+1}}Z^j_{k+1}$
    Compute $\hat\ell^b_n(\theta^n)$ approximating (8)
    $\theta^n \leftarrow$ Gradient Step$(\hat\ell^b_n(\theta^n))$
  end while
  while not converged **do**
    Sample $\{X^j_k\}^{N,M}_{k,j=0}, \{Y^j\}^M_{j=0}$ where $X^j_N \sim p_{\text{ref}}, Y^j \sim p_{\text{obs}}$, and $X^j_k = \mathbf{B}^{Y^j}_{\theta^n}(k+1, X^j_{k+1}) + \sqrt{2\gamma_{k+1}}\tilde Z^j_{k+1}$
    Compute $\hat\ell^f_{n+1}(\phi^{n+1})$ approximating (9)
    $\phi^{n+1} \leftarrow$ Gradient Step$(\hat\ell^f_{n+1}(\phi^{n+1}))$
  end while
end for
Output: (θ^L, ϕ^{L+1})

4 CDSB IMPROVEMENTS

4.1 CONDITIONAL REFERENCE MEASURE

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

As a simple example, for image super-resolution we would like to obtain high-resolution 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, the low-resolution image y itself can serve as an approximate initialization for sampling X_N. A simple model for $p_{\text{ref}}(x|y)$ is to take $p_{\text{ref}}(x|y) = \mathcal{N}(x; y, \sigma_{\text{ref}}^2 \operatorname{Id})$. We can further take $\sigma_{\text{ref}}^2 = \rho\sigma_{x|y}^2$, where ρ is a variance inflation parameter and $\sigma_{x|y}^2$ is an estimate of the conditional variance of X given Y. Equivalently, $\sigma_{x|y}^2$ is the variance of $X - Y$ which can be estimated using training data. We will also explore other classes of conditional models for $p_{\text{ref}}(x|y)$ in the experiments, such as neural network models and the Ensemble Kalman Filter.

4.2 CONDITIONAL FORWARD PROCESS

To accelerate the convergence of the IPF procedure, 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 SB solution $\pi^\star$.

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

$$\mathbb{E}[\mathrm{KL}(\pi_{Y,0}^{c,n}|p(\cdot|Y)) + \mathrm{KL}(\pi_{Y,N}^{c,n}|p_{\text{ref}})] \leq \frac{2}{n}\mathbb{E}[\mathrm{KL}\left(\pi_Y^{c,\star}|p_Y\right)],$$

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

Furthermore, if $p_{y,N} = p_{\text{ref}}(\cdot|y)$, we have that $\bar{\pi}^\star = \bar{p}$. 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 leverage the Langevin dynamics and define $p_y(x_{0:N})$ as following an unadjusted Langevin dynamics; i.e. $p_{k+1|k}(x'|x, y) = \mathcal{N}(x'; x + \gamma_{k+1}\nabla \log p_{\text{ref}}(x|y), 2\gamma_{k+1} \operatorname{Id})$. In particular, when $p_{\text{ref}}(x|y) = \mathcal{N}(x; \mu(y), \sigma^2(y) \operatorname{Id})$, this reduces to a discretized Ornstein–Uhlenbeck process. Such a process admits $p_{\text{ref}}(x|y)$ as limiting distribution as $\gamma \to 0$ and $N \to \infty$ [Durmus and Moulines, 2017].

4.3 FORWARD-BACKWARD SAMPLING

When we use $p_{\text{ref}}(x|y) = p_{\text{ref}}(x)$, our proposed method also shares connections with the transport maps techniques developed by Marzouk et al. [2016], Spantini et al. [2022]. They propose methods to learn a deterministic invertible transport map $\mathcal{S}(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ 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 $X^{\text{pos}} = \mathcal{S}(\cdot, y^{\text{obs}})^{-1}(X^{\text{ref}})$. The SB version of this approach is the intractable problem (5).

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

We can also exploit an analogous forward-backward sampling scheme in our framework $\hat{X}_N \sim \bar{p}^L(x_N),\ \ \hat{X}_0 \sim \bar{q}^L(x_0|\hat{X}_N, y^{\text{obs}})$. Since $\bar{q}^L$ is the approximate time-reversal of $\bar{p}^L$, this strategy shares similar advantages as the method of Spantini et al. [2022] when the half-bridge $\bar{p}^L(x_{0:N}|y^{\text{obs}})$ does not solve the CSB problem exactly. We call this extension CDSB-Forward/Backward (CDSB-FB).

5 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, Hermans et al., 2020]. However, these methods require knowing the prior while CDSB only needs to have access to samples. For tasks such as image inpainting, the prior is indeed implicit. While it may be evaluated approximately pointwise using the probability flow method [Song et al., 2021b, Section D], this is computationally intensive.

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 distribu-

tion 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_{join} and p_{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 is an alternative building stochastic transport maps. It appears competitive in our experiments and is a natural extension of CSGMs which provide SOTA results in a variety of domains. Training for CDSB (and CSGMs) is also more stable than for GANs as one does not have to solve an adversarial problem.

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, $p_{\text{ref}}(x) = \mathcal{N}(x; 0, \sigma_{\text{ref}}^2 \,\text{Id})$ 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 4.1). Finally, for linear Gaussian inverse problems, Kadkhodaie and Simoncelli [2021], Kawar et al. [2021, 2022] develop efficient methodologies when the Gaussian likelihood is 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. Jolicoeur-Martineau et al. [2021] introduce better SDE integrators, 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 most of these techniques are complementary to and can be readily applied in the Schrödinger bridge setting; *e.g.*one could distill the last CDSB network $\mathbf{B}^y_{\theta^L}$. Additionally, SB and CSB provide a theoretically grounded framework to perform few-step sampling.

6 EXPERIMENTS

We summarize in this section the key findings from our experiments. Further experimental details and results are included in the supplementary.

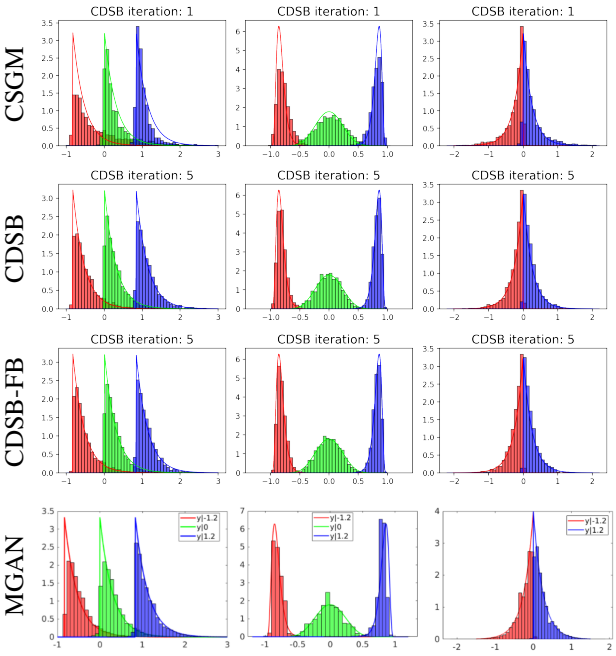


Figure 1: True posterior $p(x|y^{\text{obs}})$ for $y^{\text{obs}} \in \{-1.2, 0, 1.2\}$ (solid lines) and approximations for the 2D examples.

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

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

6.1 2D SYNTHETIC EXAMPLES

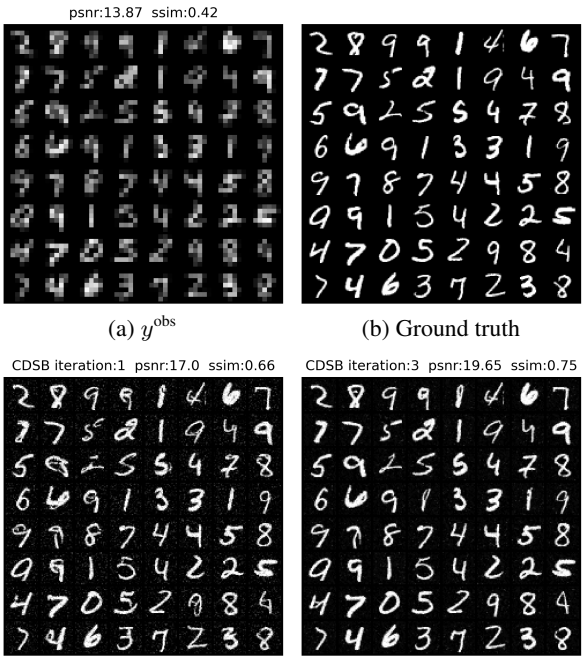
We first demonstrate the validity and accuracy of our method using the two-dimensional examples of Kovachki et al. [2021]. We consider three nonlinear, non-Gaussian examples for $p_{\text{join}}(x, y)$: define $p_{\text{obs}}(y) = \text{Unif}(y; [-3, 3])$ for all examples and $p(x|y)$ is defined through

$$\begin{aligned}
&\text{Example 1: } X = \tanh(Y) + Z, \quad Z \sim \Gamma(1, 0.3),\\
&\text{Example 2: } X = \tanh(Y + Z), \quad Z \sim \mathcal{N}(0, 0.05),\\
&\text{Example 3: } X = Z \tanh(Y), \quad Z \sim \Gamma(1, 0.3).
\end{aligned}$$

We run CDSB on each of the examples with 50,000 training points and compare with the Monotone GAN (MGAN) algorithm [Kovachki et al., 2021]. CDSB uses a neural network model with 32k parameters (approximately 6x less parameters than MGAN) with $N = 50$ diffusion steps. Figure 1 shows the resulting histogram of the learned $p(x|y^{\text{obs}})$ and the true posterior for $y^{\text{obs}} \in \{-1.2, 0, 1.2\}$. As can be

PSNR↑/SSIM↑(/FID↓)	$N = 5$	$N = 10$	$N = 10$	$N = 20$	$N = 20$	$N = 50$
CSGM	17.22/0.67	20.03/0.79	14.77/0.60	16.31/**0.71**	19.52/0.47/92.02	20.52/0.57/48.68
CDSB	18.55/**0.75**	20.69/0.79	16.24/0.62	16.61/0.66	19.72/0.50/57.22	20.70/0.59/40.08
CDSB-Cond	**19.67/0.75**	**20.95/0.84**	**16.61/0.66**	**16.71**/0.68	**21.11/0.61/28.41**	**21.46/0.65/13.71**
	(a)		(b)		(c)	

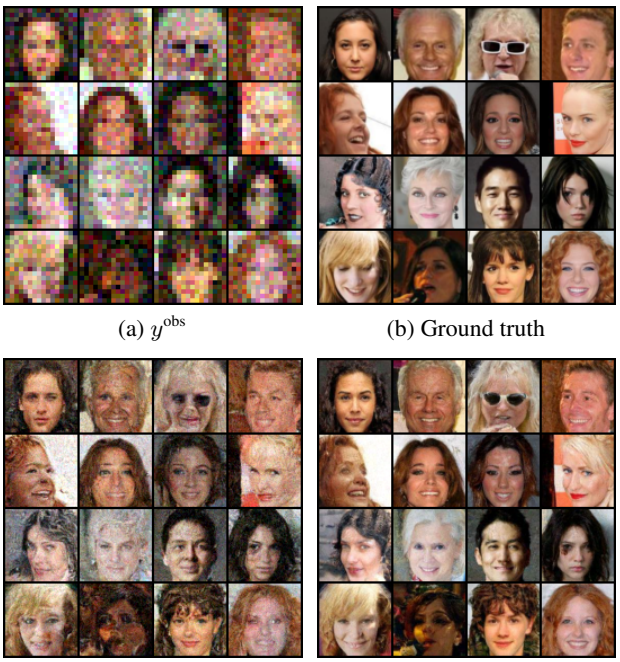
Table 2: Quantitative results for (a) MNIST 4x super-resolution; (b) MNIST 14x14 inpainting; (c) CelebA 4x super-resolution with Gaussian noise.



(a) y^{obs} (b) Ground truth

(c) CSGM (d) CDSB-Cond

Figure 2: Uncurated samples for the MNIST 4x super-resolution task with $N = 5$.



(a) y^{obs} (b) Ground truth

(c) CSGM (d) CDSB-Cond

Figure 3: Uncurated samples for the CelebA 4x super-resolution with Gaussian noise task with $N = 20$.

observed, the empirical density of CDSB samples is sharper and aligns more closely with the ground truth density. We also observe that using more CDSB iterations corrects the sampling bias compared to using only one CDSB iteration (which corresponds to CSGM). Using forward-backward sampling (CDSB-FB) further improves the sample quality.

6.2 BIOCHEMICAL OXYGEN DEMAND MODEL

We now consider a Bayesian inference problem on a biochemical oxygen demand (BOD) from Marzouk et al. [2016]. Let $X_1, X_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$, $A = 0.8 + 0.4\,\text{erf}(X_1/\sqrt{2})$, $B = 0.16 + 0.15\,\text{erf}(X_2/\sqrt{2})$ and $Y = \{Y(t)\}_{t=1}^5$ satisfy $Y(t) = A(1 - \exp(-Bt)) + Z$ with $Z \sim \mathcal{N}(0, 10^{-3})$. 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 6×10^6 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].

6.3 IMAGE EXPERIMENTS

We now apply CDSB to a range of inverse problems on image datasets. We consider the following tasks: (a) MNIST 4x super-resolution (7x7 to 28x28), (b) MNIST 14x14 inpainting, (c) CelebA 4x super-resolution (16x16 to 64x64) with Gaussian noise of $\sigma_y = 0.1$. For CDSB-Cond, we consider the following choices for conditional initialization: for

tasks (a) and (c), we use the upsampled y as described in Section 4.1; for task (b), we use a separate neural network regressor with the same architecture as $\mathbf{F}, \mathbf{B}$ to output the initialization mean. We report PSNR and SSIM (the higher the better), as well as FID scores (the lower the better) for RGB images in Table 2. We display a visual comparison between the methods in Figure 2, 3, and additional image samples in the supplementary. CDSB and CDSB-Cond 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.

6.4 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 so-called filtering distribution $p(x_t|y^{\text{obs}}_{1:t})$, that is the posterior of X_t given the observations $Y_{1:t} = y^{\text{obs}}_{1:t}$. 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] or the Ensemble Kalman filter (EnKF) [Law et al., 2015].

Assume at time t, one has a collection of samples $\{X^i_t\}_{i=1}^M$ distributed (approximately) according to $p(x_t|y^{\text{obs}}_{1:t})$. We sample $X^i_{t+1} \sim f(x_{t+1}|X^i_t)$ and $Y^i_{t+1} \sim g(y_{t+1}|X^i_{t+1})$. The resulting samples $\{X^i_{t+1}, Y^i_{t+1}\}_{i=1}^M$ are thus distributed according to $p_{\text{join}}(x_{t+1}, y_{t+1}) := p(x_{t+1}, y_{t+1}|y^{\text{obs}}_{1:t})$. We can also easily obtain samples from $p_{\text{jref}}(x_{t+1}, y_{t+1}) := p_{\text{ref}}(x_{t+1}|y_{t+1}, y^{\text{obs}}_{1:t})p(y_{t+1}|y^{\text{obs}}_{1:t})$ where $p_{\text{ref}}(x_{t+1}|y_{t+1}, y^{\text{obs}}_{1:t})$ is an easy-to-sample distribution designed by the user. Thus we can use CDSB to obtain a (stochastic) transport map between $p_{\text{join}}(x_{t+1}, y_{t+1})$ and $p_{\text{jref}}(x_{t+1}, y_{t+1})$ and applying it to $Y_{t+1} = y^{\text{obs}}_{t+1}$, we can obtain new samples from $p(x_{t+1}|y^{\text{obs}}_{1:t+1})$. A similar strategy for filtering based on deterministic transport maps was recently proposed by Spantini et al. [2022].

We apply CSGM and CDSB to the Lorenz-63 model [Law et al., 2015] following the procedure above for a time series of length 2000. We consider a short diffusion process with $N = 20$ steps, as well as a long one with $N = 100$. To accelerate the sequential inference process, in this example we use analytic basis regression instead of neural networks for all methods, and we only run 5 iterations of CDSB. We further test CDSB with conditional initialization (CDSB-Cond) as described in Section 4.1. As the EnKF is applicable to this model, we can use the resulting approximate Gaussian filtering distribution it outputs for $p_{\text{ref}}(x_{t+1}|y_{t+1}, y^{\text{obs}}_{1:t})$.

Table 3 shows that for $N = 20$ both CDSB and CDSB-Cond successfully perform filtering and outperform the EnKF, whereas CSGM fails to track the state accurately after a few hundred times steps. Furthermore, CDSB-Cond achieves the lowest error consistently.

When using $N = 100$, CSGM can achieve RMSE comparable with CDSB-Cond using $N = 20$, but CDSB and CDSB-Cond still provide advantages compared to CSGM, CDSB-Cond achieving again the lowest RMSE. 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.

M	500	1000	2000
EnKF	.354±0.006	.355±.005	.354±.003
CSGM (short)	Diverges		
CDSB (short)	.251±.011	.218±.008	.196±.005
CDSB-Cond (short)	**.236±.012**	**.207±.014**	**.178±.007**
CSGM (long)	.232±.008	.203±.009	.182±.009
CDSB (long)	.220±.012	.195±.007	.166±.004
CDSB-Cond (long)	**.218±.014**	**.185±.008**	**.160±.003**

Table 3: RMSEs over 10 runs between each algorithm's filtering means and the ground truth filtering means for $N = 20$ (short) and $N = 100$ (long). The lowest errors are highlighted in bold.

7 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. CDSB improves sample quality compared to CSGM for small number N of diffusion steps. This theoretically grounded approach is complementary to the many other techniques that have been recently proposed to accelerate SGMs, *e.g.* Dockhorn et al. [2022], Salimans and Ho [2022], Watson et al. [2022], 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^{obs} when we want to sample from $p(x|y^{\text{obs}})$ but not at the training stage. Hence if y^{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^{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.

References

Ricardo Baptista, Olivier Zahm, and Youssef Marzouk. An adaptive transport framework for joint and conditional density estimation. *arXiv preprint arXiv:2009.10303*, 2020.

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Mark A Beaumont. Approximate Bayesian computation. *Annual Review of Statistics and Its Applications*, 6:379–403, 2019.

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E Jacob. Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*, 2019.

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021a.

Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022.

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2021b.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. *arXiv preprint arXiv:2112.05146*, 2021.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2021.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GAN on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*, 2022.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656-704):3, 2009.

Alain Durmus and Éric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pages 4239–4248, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Zahra Kadkhodaie and Eero P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Advances in Neural Information Processing Systems*, 2021.

Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. In *Advances in Neural Information Processing Systems*, 2021.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021.

Nikola Kovachki, Ricardo Baptista, Bamdad Hosseini, and Youssef Marzouk. Conditional sampling with monotone GANs. *arXiv preprint arXiv:2006.06755*, 2021.

Jakob Kruse, Gianluca Detommaso, Ullrich Köthe, and Robert Scheichl. HINT: Hierarchical invertible neural transport for density estimation and Bayesian inference. In *AAAI Conference on Artificial Intelligence*, 2021.

Solomon Kullback. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39 (4):1236–1243, 1968.

Solomon Kullback. *Information Theory and Statistics*. Dover Publications, Inc., Mineola, NY, 1997. Reprint of the second (1968) edition.

Kody Law, Andrew Stuart, and Kostantinos Zygalakis. *Data Assimilation*. Springer, 2015.

Flavien Léger. A gradient descent perspective on Sinkhorn. *Applied Mathematics & Optimization*, 84(2):1843–1855, 2021.

Christian Léonard. Some properties of path measures. In *Séminaire de Probabilités XLVI*, pages 207–230. Springer, 2014a.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574, 2014b.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, pages 1–41, 2016.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

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

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020.

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.

Alessio Spantini, Ricardo Baptista, and Youssef Marzouk. Coupling techniques for nonlinear ensemble filtering. *SIAM Review*, 2022. to appear.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, 2021.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving Shrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23 (7):1661–1674, 2011.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. *arXiv preprint arXiv:2202.05830*, 2022.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv preprint arXiv:2112.07804*, 2021.

Xingyu Zhou, Yuling Jiao, Jin Liu, and Jian Huang. A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, 2022. to appear.

A ORGANIZATION OF THE SUPPLEMENTARY

The supplementary is organized as follows. We recall the DSB algorithm for unconditional simulation from De Bortoli et al. [2021] in Appendix B. The proofs of our propositions are given in Appendix C. In Appendix D, we give details on the loss functions we use to train CDSB. A continuous-time version of the conditional time-reversal and conditional DSB is presented in Appendix E. The forward-backward technique used in our experiments is detailed in Appendix F. Finally, we provide experimental details and guidelines in Appendix G.

B DIFFUSION SCHRÖDINGER BRIDGE

We recall here the DSB algorithm introduced by De Bortoli et al. [2021] which is a numerical approximation of IPF[2]. DSB can be thought of as a general This returns an approximation to the (standard unconditional) SB problem.

Algorithm 2 Diffusion Schrödinger Bridge [De Bortoli et al., 2021]

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_0^j \sim p_{\text{data}}$, and
    $X_{k+1}^j = \mathbf{F}_{\phi^n}(k, X_k^j) + \sqrt{2\gamma_{k+1}} Z_{k+1}^j$
    Compute $\hat{\ell}_n^b(\theta^n)$ approximating (10)
    $\theta^n \leftarrow$ Gradient Step$(\hat{\ell}_n^b(\theta^n))$
  end while
  while not converged **do**
    Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_N^j \sim p_{\text{ref}}$, and
    $X_{k-1}^j = \mathbf{B}_{\theta^n}(k, X_k^j, Y^j) + \sqrt{2\gamma_k} \tilde{Z}_k^j$
    Compute $\hat{\ell}_{n+1}^f(\phi^{n+1})$ approximating (11)
    $\phi^{n+1} \leftarrow$ Gradient Step$(\hat{\ell}_{n+1}^f(\phi^{n+1}))$
  end while
end for
Output: (θ^L, ϕ^{L+1})

In this (unconditional) SB scenario, the transition kernels satisfy $q_{k|k+1}^n(x|x') = \mathcal{N}(x; \mathbf{B}_{\theta^n}(k+1, x'), 2\gamma_{k+1}\,\mathrm{Id})$ and $p_{k+1|k}^n(x'|x) = \mathcal{N}(x'; \mathbf{F}_{\phi^n}(k, x), 2\gamma_{k+1}\,\mathrm{Id})$ where θ^n is obtained by minimizing

$$\ell_n^b(\theta) = \mathbb{E}_{p^n}[\textstyle\sum_k \|\mathbf{B}_\theta(k+1, X_{k+1}) - G_{n,k}(X_k, X_{k+1})\|^2] \tag{10}$$

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

$$\ell_{n+1}^f(\phi) = \mathbb{E}_{q^n}[\textstyle\sum_k \|\mathbf{F}_\phi(k, X_k) - H_{n,k}^Y(X_k, X_{k+1})\|^2] \tag{11}$$

for $H_{n,k}(x, x') = x + \mathbf{B}_{\theta^n}(k+1, x') - \mathbf{B}_{\theta^n}(k+1, x)$. See De Bortoli et al. [2021] for a derivation of these loss functions.

C PROOFS OF PROPOSITIONS

C.1 PROOF OF PROPOSITION 1

Let $\bar{\pi}$ such that $\mathrm{KL}(\bar{\pi}|\bar{p}) < +\infty$, which exists since we have that $\mathrm{KL}(\bar{\pi}^\star|\bar{p}) < +\infty$, and $\bar{\pi}_0 = p_{\text{join}},\ \bar{\pi}_N = p_{\text{jref}}$, where we define $\bar{p}(x_{0:N}, y_{0:N}) := p_{y_0}(x_{0:N})\bar{p}_{\text{obs}}(y_{0:N})$ and $\bar{p}_{\text{obs}}(y_{0:N}) := p_{\text{obs}}(y_0)\prod_{k=0}^{N-1}\delta_{y_k}(y_{k+1})$. Since $\mathrm{KL}(\bar{\pi}|\bar{p}) < +\infty$ we have using the transfer theorem [Kullback, 1997, Theorem 2.4.1] that $\mathrm{KL}(\bar{\pi}_{\text{obs}}|\bar{p}_{\text{obs}}) < +\infty$, where $\bar{\pi}_{\text{obs}}(y_{0:N}) :=$

[2]For discrete measures, IPF is also known as the Sinkhorn algorithm and can be implemented exactly [Peyré and Cuturi, 2019].

$\int_{(\mathbb{R}^d)^N} \bar{\pi}(x_{0:N}, y_{0:N})\mathrm{d}x_{0:N}$. In addition, using the chain rule for the Kullback–Leibler divergence, see [Léonard, 2014a, Theorem 2.4], we get that

$$\mathrm{KL}(\bar{\pi}_{\mathrm{obs}}|\bar{p}_{\mathrm{obs}}) = \mathrm{KL}(\bar{\pi}_{\mathrm{obs},0}|p_{\mathrm{obs}}) + \int_{\mathcal{Y}} \mathrm{KL}(\bar{\pi}_{\mathrm{obs}|0}|\bar{p}_{\mathrm{obs}|0})p_{\mathrm{obs}}(y)\mathrm{d}y < +\infty,$$

where $\bar{p}_{\mathrm{obs}|0} = \prod_{k=0}^{N-1} \delta_{y_k}(y_{k+1})$ and therefore $\bar{\pi}_{\mathrm{obs}|0} = \bar{p}_{\mathrm{obs}|0}$. Since we also have that $\pi_{\mathrm{obs},0} = p_{\mathrm{obs}}$ we get that $\pi_{\mathrm{obs}} = \bar{p}_{\mathrm{obs}}$. Hence, letting π^c be the kernel such that $\bar{\pi} = \pi^c \otimes \bar{p}_{\mathrm{obs}}$ we have using [Léonard, 2014a, Theorem 2.4] that

$$\mathrm{KL}(\bar{\pi}|\bar{p}) = \int_{\mathcal{Y}} \mathrm{KL}\left(\pi^c_y|p_y\right) p_{\mathrm{obs}}(y)\mathrm{d}y. \tag{12}$$

In addition, we have $\bar{\pi}_0 = \pi^c_0 \otimes p_{\mathrm{obs}} = p_{\mathrm{joint}}$. Similarly, we have $\bar{\pi}_N = \pi^c_N \otimes p_{\mathrm{obs}} = p_{\mathrm{jref}}$. Hence, $\pi^c_{y,0} = p(\cdot|y)$ and $\pi^c_N = p_{\mathrm{ref}}$, p_{obs}-almost surely. Let $\bar{\pi}^\star = \pi^{\star,c} \otimes \bar{p}_{\mathrm{obs}}$ be the minimizer of (7) and $\hat{\pi}^c$ be the minimizer of (6). Then, we have that $\bar{\pi} = \hat{\pi}^c \otimes \bar{p}_{\mathrm{obs}}$ satisfies $\mathrm{KL}(\bar{\pi}^\star|\bar{p}) \leq \mathrm{KL}(\bar{\pi}|\bar{p})$. Using (12), we have that $\mathbb{E}[\mathrm{KL}\left(\pi^{\star,c}_Y|p_Y\right)] \leq \mathbb{E}[\mathrm{KL}\left(\hat{\pi}^c_Y|p_Y\right)]$. But we have that $\mathbb{E}[\mathrm{KL}\left(\hat{\pi}^c_Y|p_Y\right)] \leq \mathbb{E}[\mathrm{KL}\left(\pi^{\star,c}_Y|p_Y\right)]$ since $\hat{\pi}^c$ is the minimizer of (6). Using the uniqueness of the minimizer of (6) we have that $\pi^{\star,c} = \hat{\pi}^c$, which concludes the proof.

C.2 PROOF OF PROPOSITION 2

Let $n \in \mathbb{N}$. and $\bar{q}$ such that $\mathrm{KL}(\bar{q}|\bar{p}^n) < +\infty$ and $\bar{q} = p_{\mathrm{jref}}$ (note that the existence of such a distribution is ensured since $\mathrm{KL}(p_{\mathrm{join}} \otimes p_{\mathrm{jref}}|\bar{p}_{0,N}) < +\infty$). Using the chain rule for the Kullback–Leibler divergence, see [Léonard, 2014a, Theorem 2], we have

$$\mathrm{KL}(\bar{q}|\bar{p}^n) = \mathrm{KL}\left(\bar{q}_{\mathrm{obs}}|\bar{p}_{\mathrm{obs}}\right) + \int_{\mathcal{Y}^{N+1}} \mathrm{KL}(\bar{q}_{|\mathrm{obs}}|\bar{p}^n_{|\mathrm{obs}})\mathrm{d}\bar{q}_{\mathrm{obs}}(y_{0:N}), \tag{13}$$

where $\bar{q}_{\mathrm{obs}} = \int_{\mathcal{X}^{N+1}} \bar{q}(x_{0:N}, y_{0:N})\mathrm{d}x_{0:N}$ and $\bar{q}_{|\mathrm{obs}}$ and $\bar{p}^n_{|\mathrm{obs}}$ are the conditional distribution of $\bar{q}$, respectively $\bar{p}^n$ w.r.t to $y_{0:N}$. Since $\mathrm{KL}\left(\bar{q}_{\mathrm{obs}}|\bar{p}_{\mathrm{obs}}\right) < +\infty$, we can use [Léonard, 2014a, Theorem 2.4] and we have

$$\mathrm{KL}\left(\bar{q}_{\mathrm{obs}}|\bar{p}_{\mathrm{obs}}\right) = \mathrm{KL}\left(\bar{q}_{\mathrm{obs},N}|\bar{p}_{\mathrm{obs},N}\right) + \int_{\mathcal{Y}} \mathrm{KL}\left(\bar{q}_{\mathrm{obs}|N}|\bar{p}_{\mathrm{obs}|N}\right) \mathrm{d}\bar{q}_{\mathrm{obs},N}(y_N),$$

with $\bar{p}^n_{\mathrm{obs}|N}(y_{0:N-1}|y_N) = \prod_{k=0}^{N-1} \delta_{y_{k+1}}(y_k)$. Therefore, since $\mathrm{KL}\left(\bar{q}_{\mathrm{obs}}|\bar{p}_{\mathrm{obs}}\right) < +\infty$, we get that $\bar{q}^n_{\mathrm{obs}|N}(y_{0:N-1}|y_N) = \prod_{k=0}^{N-1} \delta_{y_{k+1}}(y_k)$. Since $\bar{q}_{\mathrm{obs},N} = p_{\mathrm{obs}}$, we get that $\bar{q}(x_{0:N}, y_{0:N}) = \bar{p}_{\mathrm{obs}}(y_{0:N})\bar{q}^n(x_{0:N}|y_{0:N}) = \bar{p}_{\mathrm{obs}}(y_{0:N})\bar{q}^n(x_{0:N}|y_N)$, where we have used that $y_N = y_k$ for $k \in \{0, \ldots, N\}$, $\bar{p}_{\mathrm{obs}}(y_{0:N})$ almost surely. Combining this result and (13) we get that

$$\mathrm{KL}(\bar{q}|\bar{p}^n) = \int_{\mathcal{Y}^{N+1}} \mathrm{KL}(\bar{q}_{|\mathrm{obs}}|\bar{p}^n_{|\mathrm{obs}})\mathrm{d}p_{\mathrm{obs}}(y_{0:N}) = \int_{\mathcal{Y}} \mathrm{KL}(\bar{q}(\cdot|y_N)|\bar{p}^n(\cdot|y_N))\mathrm{d}p_{\mathrm{obs}}(y_N),$$

Using [Léonard, 2014a, Theorem 2], we have that for any $y_N \in \mathcal{Y}$

$$\mathrm{KL}(\bar{q}(\cdot|y_N)|\bar{p}^n(\cdot|y_N) = \mathrm{KL}(p_{\mathrm{ref}}|\bar{p}^n_N(\cdot|y_N)) + \int_{\mathcal{Y}} \mathrm{KL}(\bar{q}(\cdot|y_N, x_N)|\bar{p}^n(\cdot|y_N, x_N))p_{\mathrm{ref}}(x_N)\mathrm{d}x_N.$$

We get that $\bar{q}^n(\cdot|y_N, x_N) = \bar{p}^n(\cdot|y_N, x_N)$. Therefore for any $x_{0:N} \in \mathcal{X}^{N+1}$ and $y_N \in \mathcal{Y}$,

$$\bar{q}^n(x_{0:N}|y_N) = p_{\mathrm{ref}}(x_N) \prod_{k=0}^{N-1} \bar{p}^n_{k|k+1}(x_k|x_{k+1}, y_N).$$

The proof that for any $x_{0:N} \in \mathcal{X}^{N+1}$ and $y_0 \in \mathcal{Y}$, we have

$$\bar{p}^{n+1}(x_{0:N}|y_0) = p(x_0|y_0) \prod_{k=0}^{N-1} \bar{q}^n_{k+1|k}(x_{k+1}|x_k, y_0),$$

is similar.

C.3 PROOF OF PROPOSITION 3

Using [Léger, 2021, Corollary 1], we get that for any $n \in \mathbb{N}$ with $n \geq 1$

$$\mathrm{KL}(\bar{\pi}^n_0|p_{\mathrm{join}}) + \mathrm{KL}(\bar{\pi}^n_N|p_{\mathrm{jref}}) \leq \frac{2}{n}\,\mathrm{KL}(\bar{\pi}^\star|\bar{p}). \tag{14}$$

Similarly to Proposition 2, we have that for any $n \in \mathbb{N}$, there exists a Markov kernel $\pi^{c,n}$ such that $\bar{\pi}^n = \bar{p}_{\mathrm{obs}} \otimes \pi^{c,n}$. Recall that there exists a Markov kernel $\pi^{c,\star}$ such that $\bar{\pi}^\star = \bar{p}_{\mathrm{obs}} \otimes \pi^{c,\star}$ and that $\bar{p} = \bar{p}_{\mathrm{obs}} \otimes p_y$. Hence, using [Léonard, 2014a, Theorem 2.4], we get that for any $n \in \mathbb{N}$,

$$\mathrm{KL}(\bar{\pi}^n_0|p_{\mathrm{join}}) = \mathbb{E}[\mathrm{KL}(\pi^{c,n}_{Y,0}|p(\cdot|Y))], \qquad \mathrm{KL}(\bar{\pi}^n_N|p_{\mathrm{jref}}) = \mathbb{E}[\mathrm{KL}(\pi^{c,n}_{Y,N}|p_{\mathrm{ref}})]. \tag{15}$$

Similarly, we have that

$$\mathrm{KL}(\bar{\pi}^\star|\bar{p}) = \mathbb{E}[\mathrm{KL}\left(\pi^{c,\star}_Y|p_Y\right)]. \tag{16}$$

We conclude the proof upon combining (14), (15) and (16).

D DETAILS ON THE LOSS FUNCTIONS

In this section, 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 $\bar{p}^n$ and $\bar{q}^n$, similarly to Section 3.2. In Section 3.2, the transitions satisfy $\bar{q}^n_{k|k+1}(x|x', y) = \mathcal{N}(x; \mathbf{B}^y_{\theta^n}(k+1, x'), 2\gamma_{k+1}\,\mathrm{Id})$ and $\bar{p}^n_{k+1|k}(x'|x, y) = \mathcal{N}(x'; \mathbf{F}^y_{\phi^n}(k, x), 2\gamma_{k+1}\,\mathrm{Id})$ where θ^n is obtained by minimizing

$$\ell^b_n(\theta) = \mathbb{E}_{\bar{p}^n}[\textstyle\sum_k \|\mathbf{B}^Y_\theta(k+1, X_{k+1}) - G^Y_{n,k}(X_k, X_{k+1})\|^2]$$

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

$$\ell^f_{n+1}(\phi) = \mathbb{E}_{\bar{q}^n}[\textstyle\sum_k \|\mathbf{F}^Y_\phi(k, X_k, Y) - H^Y_{n,k}(X_k, X_{k+1})\|^2]$$

for $H^y_{n,k}(x, x') = x + \mathbf{B}^y_{\theta^n}(k+1, x') - \mathbf{B}^y_{\theta^n}(k+1, x)$. We justify these formulas by proving the following result which is a straightforward extension of De Bortoli et al. [2021]. We recall that for any $n \in \mathbb{N}$, $k \in \{0, \dots, N\}$, $x_k, x_{k+1} \in \mathbb{R}^d$ and $y \in \mathcal{Y}$, $b^{n,y}_{k+1}(x_{k+1}) = -f^{n,y}_k(x_{k+1}) + 2\nabla \log \bar{p}^n_{k+1}(x_{k+1}|y)$ and $f^{n+1,y}_k(x_k) = -b^{n,y}_{k+1}(x_k) + 2\nabla \log \bar{q}^n_k(x_k|y)$[3].

Proposition 4. *Assume that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$, $\bar{q}_k(\cdot|y)$ and $\bar{p}_k(\cdot|y)$ are bounded and*

$$\bar{q}^n_{k|k+1}(x_k|x_{k+1}, y) = \mathcal{N}(x_k; B^{n,y}_{k+1}(x_{k+1}), 2\gamma_{k+1}\,\mathrm{Id}),\ \bar{p}^n_{k+1|k}(x_{k+1}|x_k, y) = \mathcal{N}(x_{k+1}; F^{n,y}_k(x_k), 2\gamma_{k+1}\,\mathrm{Id}),$$

with $B^{n,y}_{k+1}(x) = x + \gamma_{k+1} b^{n,y}_{k+1}(x)$, $F^{n,y}_k(x) = x + \gamma_{k+1} f^{n,y}_k(x)$ for any $x \in \mathbb{R}^d$. Then we have for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d \times \mathcal{Y}, \mathbb{R}^d)} \mathbb{E}_{\bar{p}^n}[\|\mathrm{B}(X_{k+1}, Y) - G^Y_{n,k}(X_k, X_{k+1})\|^2], \tag{17}$$

$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d \times \mathcal{Y}, \mathbb{R}^d)} \mathbb{E}_{\bar{q}^n}[\|\mathrm{F}(X_k, Y) - H^Y_{n,k}(X_k, X_{k+1})\|^2], \tag{18}$$

$$G^y_n(x, x') = x' + F^{n,y}_k(x) - F^{n,y}_k(x'), \qquad H^y_n(x, x') = x + B^{n,y}_{k+1}(x') - B^{n,y}_{k+1}(x).$$

Proof. We only prove (17) since the proof (18) is similar. Let $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. For any $x_{k+1} \in \mathbb{R}^d$ we have

$$\bar{p}^n_{k+1}(x_{k+1}|y) = (4\pi\gamma_{k+1})^{-d/2} \textstyle\int_{\mathbb{R}^d} \bar{p}^n(x_k|y) \exp[-\|F^{n,y}_k(x_k) - x_{k+1}\|^2/(4\gamma_{k+1})] \mathrm{d}x_k,$$

with $F^{n,y}_k(x_k) = x_k + \gamma_{k+1} f^{n,y}_k(x_k)$. Since $\bar{p}^n_k > 0$ is bounded using the dominated convergence theorem we have for any $x_{k+1} \in \mathbb{R}^d$

$$\nabla_{x_{k+1}} \log \bar{p}^n_{k+1}(x_{k+1}|y) = \textstyle\int_{\mathbb{R}^d} (F^{n,y}_k(x_k) - x_{k+1})/(2\gamma_{k+1})\, \bar{p}_{k|k+1}(x_k|x_{k+1}, y) \mathrm{d}x_k.$$

Therefore we get that for any $x_{k+1} \in \mathbb{R}^d$

$$b^{n,y}_{k+1}(x_{k+1}) = \textstyle\int_{\mathbb{R}^d} (F^{n,y}_k(x_k) - F^{n,y}_k(x_{k+1}))/\gamma_{k+1}\, \bar{p}_{k|k+1}(x_k|x_{k+1}, y) \mathrm{d}x_k.$$

This is equivalent to

$$B^{n,y}_{k+1}(x_{k+1}) = \mathbb{E}[X_{k+1} + F^{n,Y}_k(X_k) - F^{n,Y}_k(X_{k+1}) | X_{k+1} = x_{k+1}, Y = y],$$

Hence, we get that

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d \times \mathcal{Y}, \mathbb{R}^d)} \mathbb{E}_{\bar{p}^n}[\|\mathrm{B}(X_{k+1}, Y) - (X_{k+1} + F^{n,Y}_k(X_k) - F^{n,Y}_k(X_{k+1}))\|^2],$$

which concludes the proof. □

E CONTINUOUS-TIME VERSIONS OF CSGM AND CDSB

In the following section, we consider the continuous-time version of CSGM and CDSB. The continuous-time dynamics we recover can be seen as the extensions of the continuous-time dynamics obtained in the unconditional setting, see Song et al. [2021b], De Bortoli et al. [2021].

[3] We should have conditioned w.r.t y_N and y_0 but since $y_0 = y_1 = \cdots = y_N$ under p_{obs} we simply conditioned by y which can be any of these values.

E.1 NOTATION

We start by introducing a few notations. The space of continuous functions from $[0,T]$ to $\mathbb{R}^d \times \mathcal{Y}$ is denoted $\mathcal{C} = \mathrm{C}([0,T], \mathbb{R}^d \times \mathcal{Y})$ and we denote $\mathscr{P}(\mathcal{C})$ the set of probability measures defined on $\mathcal{C}$. A probability measure $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is *associated with a diffusion* if it is a solution to a martingale problem, i.e. $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is associated with $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ if for any $\varphi \in \mathrm{C}_c^2(\mathbb{R}^d, \mathbb{R})$, $(\mathbf{Z}_t^\varphi)_{t\in[0,T]}$ is a $\mathbb{P}$-local martingale, where for any $t \in [0,T]$

$$\mathbf{Z}_t^\varphi = \varphi(\mathbf{X}_t) - \int_0^t \mathcal{A}_s(\varphi)(\mathbf{X}_s)\mathrm{d}s, \qquad \mathcal{A}_t(\varphi)(x) = \langle b(t,x), \nabla\varphi(x)\rangle + \Delta\varphi(x).$$

Here $\mathrm{C}_c^2(\mathbb{R}^d, \mathbb{R})$ denotes the space of twice differentiable functions from $\mathbb{R}^d$ to $\mathbb{R}$ with compact support. Doing so, $\mathbb{P}$ is uniquely defined up to the initial distribution $\mathbb{P}_0$. Finally, for any $\mathbb{P} \in \mathscr{P}(\mathcal{C})$, we introduce $\mathbb{P}^R$ the time reversal of $\mathbb{P}$, *i.e.* for any $\mathsf{A} \in \mathcal{B}(\mathcal{C})$ we have $\mathbb{P}^R(\mathsf{A}) = \mathbb{P}(\mathsf{A}^R)$ where $\mathsf{A}^R = \{t \mapsto \omega(T-t) \,:\, \omega \in \mathsf{A}\}$.

E.2 CONTINUOUS-TIME CSGM

Recall that in the unconditional setting, we consider a forward noising dynamics $(\mathbf{X}_t)_{t\in[0,T]}$ initialized with $\mathbf{X}_0 \sim p_{\text{data}}$ and satisfying the following Stochastic Differential Equation (SDE) $\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, i.e. an Ornstein–Ulhenbeck process. In this case, under entropy condition on $(\mathbf{X}_t)_{t\in[0,T]}$ (see Cattiaux et al. [2021] for instance) we have that the time-reversal process $(\tilde{\mathbf{X}}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ also satisfy an SDE given by $\mathrm{d}\tilde{\mathbf{X}}_t = \{\tilde{\mathbf{X}}_t + 2\nabla \log p_t(\tilde{\mathbf{X}}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, where p_t is the density of $\mathbf{X}_t$ w.r.t. the Lebesgue measure, and $(\tilde{\mathbf{X}}_t)_{t\in[0,T]}$ is initialized with $\tilde{\mathbf{X}}_0 \sim \mathcal{L}(\mathbf{X}_T)$, the law of $\mathbf{X}_T$ of density q_T. Using the geometric ergodicity of the Ornstein–Ulhenbeck process, $\mathcal{L}(\mathbf{X}_T)$ is close (w.r.t to the Kullback–Leibler divergence for instance) to $p_{\text{ref}} = \mathcal{N}(0, \mathrm{Id})$. Hence, we obtain that considering $(\mathbf{Z}_t)_{t\in[0,T]}$ such that $\mathbf{Z}_0 \sim \mathcal{N}(0, \mathrm{Id})$ and $\mathrm{d}\mathbf{Z}_t = \{\mathbf{Z}_t + 2\nabla \log p_t(\mathbf{Z}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, $\mathbf{Z}_T$ is approximately distributed according to p_{data}. The Euler–Maruyama discretization of $(\mathbf{Z}_t)_{t\in[0,T]}$ is the SGM used in existing work.

In the conditional setting, we consider the following dynamics $\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\mathbf{Y}_t = 0$, where $(\mathbf{X}_0, \mathbf{Y}_0) \sim p_{\text{join}}$. Note that we have $\mathbf{Y}_t = \mathbf{Y}_0$ for all $t \in [0,T]$. Using the ergodicity of the Ornstein–Ulhenbeck process, we get that $\mathcal{L}(\mathbf{X}_T, \mathbf{Y}_t)$ is close (w.r.t to the Kullback–Leibler divergence for instance) to p_{jref}. Let $(\tilde{\mathbf{X}}_t, \tilde{\mathbf{Y}}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t}, \mathbf{Y}_{T-t})_{t\in[0,T]}$. We have that $\mathrm{d}\tilde{\mathbf{X}}_t = \{\tilde{\mathbf{X}}_t + 2\nabla \log p_t(\tilde{\mathbf{X}}_t|\tilde{\mathbf{Y}}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\tilde{\mathbf{Y}}_t = 0$ with $\tilde{\mathbf{X}}_0, \tilde{\mathbf{Y}}_0 \sim \mathcal{L}(\mathbf{X}_T, \mathbf{Y}_T)$. Hence, we obtain that considering $(\mathbf{Z}_t)_{t\in[0,T]}$ such that $(\mathbf{Z}_0, \mathbf{Y}_0) \sim p_{\text{jref}}$ and $\mathrm{d}\mathbf{Z}_t = \{\mathbf{Z}_t + 2\nabla \log p_t(\mathbf{Z}_t|\mathbf{Y}_0)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, $\mathbf{Z}_T$ is approximately distributed according to p_{data}. The Euler–Maruyama discretization of $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\in[0,T]}$ is the conditional SGM.

E.3 CONNECTION WITH NORMALIZING FLOWS AND ESTIMATION OF THE EVIDENCE

It has been shown that SGMs can be used for log-likelihood computation. Here, we further show that they can be used to estimate the evidence $\log p(y^{\text{obs}})$ when $g(y^{\text{obs}}|x)$ can be computed pointwise. This is the case for many models considered in the diffusion literature, see for instance Kadkhodaie and Simoncelli [2021], Kawar et al. [2021, 2022]. Indeed, we have that for any $x \in \mathbb{R}^d$, $\log p(y^{\text{obs}}) = \log g(y^{\text{obs}}|x) + \log p(x) - \log p(x|y^{\text{obs}})$. The term $\log p(x)$ can be estimated using an unconditional SGM whereas the term $\log p(x|y^{\text{obs}})$ can be estimated using a CSGM. Note that both conditional and unconditional SGM can be trained simultaneously adding a "sink" state to $\mathcal{Y}$, i.e. considering $\mathcal{Y} \cup \{\emptyset\}$, see Ho and Salimans [2021] for instance.

We briefly explain how one can compute $\log p(x|y^{\text{obs}})$ and refer to Song et al. [2021b] for a similar discussion in the unconditional setting. Recall that the forward noising process is given by $\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\mathbf{Y}_t = 0$, where $(\mathbf{X}_0, \mathbf{Y}_0) \sim p_{\text{join}}$. We introduce another process $(\hat{\mathbf{X}}_t, \hat{\mathbf{Y}}_t)_{t\in[0,T]}$ with deterministic dynamics which has the same marginal distributions, i.e. $\mathcal{L}(\mathbf{X}_T, \mathbf{Y}_T) = \mathcal{L}(\hat{\mathbf{X}}_T, \hat{\mathbf{Y}}_T)$. This process is defined by $\mathrm{d}\hat{\mathbf{X}}_t = \{-\hat{\mathbf{X}}_t - \nabla \log p_t(\hat{\mathbf{X}}_t|\hat{\mathbf{Y}}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\hat{\mathbf{Y}}_t = 0$ with $(\hat{\mathbf{X}}_0, \hat{\mathbf{Y}}_0) \sim p_{\text{join}}$. As one has $\mathrm{d}\log p_t(\hat{\mathbf{X}}_t|\hat{\mathbf{Y}}_t) = \mathrm{div}(-\hat{\mathbf{X}}_t - \nabla \log p_t(\hat{\mathbf{X}}_t|\hat{\mathbf{Y}}_t))\mathrm{d}t$, we can approximately compute $\log p(\hat{\mathbf{X}}_0|\hat{\mathbf{Y}}_0)$ by integrating numerically this Ordinary Differential Equation (ODE). There are practically three sources of errors, one is the score approximation, one is the numerical integration error and the last one one is due to the fact that $\mathcal{L}(\hat{\mathbf{X}}_T)$ is unknown so we use the approximation $\mathcal{L}(\hat{\mathbf{X}}_T) \approx p_{\text{ref}}$.

E.4 CONTINUOUS-TIME CDSB

In this section, we introduce an IPF algorithm for solving CSB problems in continuous-time. The following results are a generalization to the conditional framework of the continuous-time results of De Bortoli et al. [2021]. The CDSB algorithm described in Algorithm 1 can be seen as a Euler–Maruyama discretization of this IPF scheme combined to neural network approximations of the drifts. Let $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ be a given reference measure (thought as the continuous time analog of $\bar{p}$). The dynamical continuous formulation of the SB problem can be written as follows

$$\Pi^\star = \arg\min \left\{ \mathrm{KL}(\Pi|\mathbb{P}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_0 = p_{\text{join}},\, \Pi_T = p_{\text{jref}} \right\}.$$

We define the IPF $(\Pi^n)_{n\in\mathbb{N}}$ such that $\Pi^0 = \mathbb{P}$ and associated with $\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\mathbf{Y}_t = 0$, with $(\mathbf{X}_0, \mathbf{Y}_0) \sim p_{\text{join}}$. Next for any $n \in \mathbb{N}$ we define

$$\Pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_T = p_{\text{jref}} \right\},$$
$$\Pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n+1}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_0 = p_{\text{join}} \right\}.$$

The following result is the continuous counterpart of Proposition 2.

Proposition 5. *Assume that $p_N, p_{\text{ref}} > 0$, $\mathrm{H}(p_{\text{ref}}) < +\infty$ and $\int_{\mathbb{R}^d} |\log p_{N|0}(x_N|x_0)| p_{\text{data}}(x_0) p_{\text{ref}}(x_N) < +\infty$. In addition, assume that there exist $\mathbb{M} \in \mathscr{P}(\mathcal{C})$, $U \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$, $C \geq 0$ such that for any $n \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\mathrm{KL}(\Pi^n|\mathbb{M}) < +\infty$, $\langle x, \nabla U(x)\rangle \geq -C(1 + \|x\|^2)$ and $\mathbb{M}$ is associated with $(\mathbf{X}_t, \mathbf{Y}_t)_{t\in[0,T]}$ such that*

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathrm{d}\mathbf{Y}_t = 0 \tag{19}$$

with $\mathbf{X}_0$ distributed according to the invariant distribution of (19)*. Then, for any $n \in \mathbb{N}$ we have:*

(a) $(\Pi^{2n+1})^R$ is associated with $(\mathbf{X}_t^{2n+1}, \mathbf{Y}_t^{2n+1})_{t\in[0,T]}$ such that $\mathrm{d}\mathbf{X}_t^{2n+1} = b_{T-t}^n(\mathbf{X}_t^{2n+1}, \mathbf{Y}_t^{2n+1})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathrm{d}\mathbf{Y}_t^{2n+1} = 0$ with $(\mathbf{X}_0^{2n+1}, \mathbf{Y}_0^{2n+1}) \sim p_{\text{jref}}$;
(b) Π^{2n+2} is associated with $\mathrm{d}\mathbf{X}_t^{2n+2} = f_t^{n+1}(\mathbf{X}_t^{2n+2}, \mathbf{Y}_t^{2n+2})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ with $(\mathbf{X}_0^{2n+2}, \mathbf{Y}_0^{2n+2}) \sim p_{\text{join}}$;
where for any $n \in \mathbb{N}$, $t \in [0,T]$, $x \in \mathbb{R}^d$ and $y \in \mathcal{Y}$, $b_t^n(x,y) = -f_t^n(x,y) + 2\nabla \log p_t^n(x|y)$, $f_t^{n+1}(x,y) = -b_t^n(x,y) + 2\nabla \log q_t^n(x|y)$, with $f_t^0(x) = -x$, and $p_t^n(\cdot|y)$, $q_t^n(\cdot|y)$ the densities of $\Pi_{t|y}^{2n}$ and $\Pi_{t|y}^{2n+1}$.

Proof. The proof of this proposition is a straightforward extension of [De Bortoli et al., 2021, Proposition 6]. □

We have seen in Appendix E.3 that it is possible to use CSGM to evaluate numerically the evidence when $g(y^{\text{obs}}|x)$ can be computed pointwise. The same strategy can be applied to both DSB and CDSB; see [De Bortoli et al., 2021, Section H.3] for details for DSB. In both cases, there exists an ordinary differential equation admitting the same marginals as the diffusion solving the SB, resp. the CSB, problem. By integrating these ODEs, we can obtain $\log p(x)$ and $\log p(x|y^{\text{obs}})$ for any x and thus can compute the evidence. Contrary to SGM and CSGM, the terminal state of the diffusion is exactly equal to the reference measure by design. So practically, we only have two instead of three sources of errors for SGM/CSGM: one is the drift approximation, one is the numerical integration error.

F FORWARD-BACKWARD SAMPLING

We detail in this section the forward-backward sampling approach and its connection with Spantini et al. [2022]. In Spantini et al. [2022], it is proposed to first learn a deterministic transport map $\mathcal{U}(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ from $(X^{\text{join}}, Y^{\text{join}}) \sim p_{\text{join}}$ to p_{jref}, then transport back the X-component through $\mathcal{S}(\cdot, y^{\text{obs}})^{-1}$ where $\mathcal{S}$ is the $\mathcal{X}$ component of $\mathcal{U}$. In other words, this is to say sampling $X^{\text{pos}} \sim p(x|y^{\text{obs}})$ corresponds to the two-step transformation

$$\hat{X}^{\text{ref}}, \hat{Y}^{\text{ref}} = \mathcal{U}(X^{\text{join}}, Y^{\text{join}}), \;\; X^{\text{pos}} = \mathcal{S}(\cdot, y^{\text{obs}})^{-1}(\hat{X}^{\text{ref}}). \tag{20}$$

The proposed CSB (7) can be thought of as the SB version of this idea. We learn a stochastic transport map from $p_{\text{join}}(x,y)$ to $p_{\text{ref}}(x,y)$. The CSB $\pi^\star$ defines, when conditioned on x_0 and y^{obs}, a (stochastic) transport map $\pi^\star_{y^{\text{obs}}}(x_N|x_0)$ from $p(x_0|y^{\text{obs}})$ to $p_{\text{ref}}(x_N|y^{\text{obs}}) = p_{\text{ref}}(x_N)$; and, when conditioned on x_N and y^{obs}, a (stochastic) transport map $\pi^\star_{y^{\text{obs}}}(x_0|x_N)$ from $p_{\text{ref}}(x_N)$ to $p(x_0|y^{\text{obs}})$. In practice, we learn using CDSB separate half-bridges $\bar{p}^L(x_{1:N}|x_0, y^{\text{obs}})$ and $\bar{q}^L(x_{0:N-1}|x_N, y^{\text{obs}})$.

Spantini et al. [2022] remarked that, since the estimator $\mathcal{S}$ may be imperfect, $\hat{X}^{\text{ref}}$ may not have distribution p_{ref} exactly. In this case, (20) allows for the cancellation of errors between $\mathcal{S}$ and $\mathcal{S}(\cdot, y^{\text{obs}})^{-1}$.

We can exploit a similar idea in the CSB framework by defining an analogous forward-backward sampling procedure

$$\hat{X}_N \sim \bar{p}^L(x_N), \quad \hat{X}_0 \sim \bar{q}^L(x_0|\hat{X}_N, y^{\text{obs}}). \tag{21}$$

As $\bar{q}^L$ is the approximate time reversal of $\bar{p}^L$, (21) exhibits similar advantages as (20) when the half-bridge $\bar{p}^L(x_{0:N}|y^{\text{obs}})$ is only an approximation to the CSB solution. While the forward and backward processes are stochastic and are not exact inverses of each other, using this forward-backward sampling may inevitably lead to increased variance. However, we found in practice that this forward-backward sampling procedure generally improves sampling quality (see *e.g.* Figures 1, 7).

G EXPERIMENTAL DETAILS

G.1 EXPERIMENTAL SETUP

Network parameterization. Two parameterizations are possible for learning $\mathbf{F}$ and $\mathbf{B}$. In the main text, we described one parameterization in which we parameterize $\mathbf{F}, \mathbf{B}$ directly as $\mathbf{F}^y_\phi(k, x), \mathbf{B}^y_\theta(k, x)$ and learn the network parameters ϕ, θ. Alternatively, we can parameterize $\mathbf{F}^y(k, x) = x + \gamma_{k+1}\mathbf{f}^y_\phi(k, x), \mathbf{B}^y(k+1, x) = x + \gamma_{k+1}\mathbf{b}^y_\theta(k+1, x)$ and learn the network parameters ϕ, θ for $\mathbf{f}^y_\phi, \mathbf{b}^y_\theta$ instead. For the 2D and BOD examples, we use a fully connected network with positional encodings as in De Bortoli et al. [2021] to learn $\mathbf{f}^y_\phi, \mathbf{b}^y_\theta$, with y as an additional input by concatenation with x. For the MNIST and CelebA examples, we follow earlier work and utilize the conditional U-Net architecture in Dhariwal and Nichol [2021]. Since residual connections are already present in the U-Net architecture, we can adopt the $\mathbf{F}^y_\phi, \mathbf{B}^y_\theta$ parameterization. In our experiments, we experiment with both parameterizations and find that the $\mathbf{f}^y_\phi, \mathbf{b}^y_\theta$ parameterization is more suitable for neural network architectures without residual connections. On the other hand, both parameterizations obtained good experimental results when using the U-Net architecture. For consistency, all reported image experiment results use the $\mathbf{F}^y_\phi, \mathbf{B}^y_\theta$ parameterization, and we leave the choice of optimal parameterization as future research.

Network warm-starting. As observed by De Bortoli et al. [2021], since the networks at IPF iteration n are close to the networks at iteration $n-1$, it is possible to warm-start ϕ^n, θ^n at ϕ^{n-1}, θ^{n-1} respectively. Empirically, we observe that this approach can significantly reduce training time at each CDSB iteration. Compared to CSGM, we usually observe immediate improvement in $\mathbf{B}_{\theta^2}$ during CDSB iteration 2 when the network is warm-started at θ^1 after CDSB iteration 1 (see *e.g.* Figure 4). As CSGM corresponds to the training objective of θ^1 at CDSB iteration 1, this shows that the CDSB framework is a generalization of CSGM with observable benefits starting CDSB iteration 2.

Conditional initialization. In the main text, we considered joint reference measures of the form $p_{\text{jref}}(x, y) = p_{\text{ref}}(x|y)p_{\text{obs}}(y)$ and simple choices for $p_{\text{ref}}(x|y)$ such as $\mathcal{N}(x; y, \sigma^2_{\text{ref}}\,\text{Id})$ for image super-resolution. We also explore two more choices for $p_{\text{ref}}(x|y)$ in our experiments. The first choice simply replaces the initialization mean from y to a neural network function $\mu_{\text{ref}}(y)$. This neural network can be pre-trained directly to estimate the conditional mean of $p(x|y)$ using standard regression with MSE loss. In the case of multi-modal $p(x|y)$ such as in the case of image inpainting, we can also train $\mu_{\text{ref}}(y)$ to estimate the conditional mean of $p_N(x_N|y)$, where x_N follows a standard diffusion process. In essence, we can train $\mu_{\text{ref}}(y)$ to facilitate $p_N(x_N|y) \approx p_{\text{ref}}(x_N|y)$ and shorten the noising process. Note that the CDSB framework is still useful in this context since $p_N(x_N|y)$ may not be well-approximated by a Gaussian distribution, which is precisely the issue CDSB is designed to tackle. Another class of conditional initialization we consider is the Ensemble Kalman Filter (EnKF), which is an ensemble-based method approximating linear Gaussian posterior updates. In this case, $p_{\text{ref}}(x|y)$ is taken to be $\mathcal{N}(x; \mu_{\text{ref}}(y), \text{diag}(\sigma^2_{\text{ref}}(y))$ where $\mu_{\text{ref}}(y), \sigma^2_{\text{ref}}(y)$ are the sample mean and variance of the EnKF posterior ensemble. Intuitively, $p_{\text{ref}}(x|y)$ is now an approximation of the true posterior $p(x|y)$ using linear prior-to-posterior mappings, which is further corrected for non-linearity and non-Gaussianity by the CDSB.

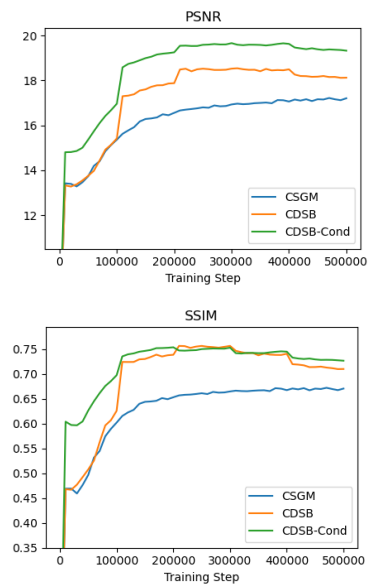


Figure 4: Test set PSNR and SSIM against the number of training steps for MNIST 4x super-resolution.

Time step schedule. For the sequence $\{\gamma_k\}_{k=1}^N$, following Ho et al. [2020], Dhari-

		MCMC	CDSB	CDSB-FB	CDSB-Cond	MGAN	IT
Mean	x_1	.075	.066±.010	.068±.010	**.072±.007**	.048	.034
	x_2	.875	.897±.019	.897±.017	**.891±.013**	.918	.902
Var	x_1	.190	.184±.007	**.190±.007**	.188±.005	.177	.206
	x_2	.397	.387±.006	.391±.006	**.393±.005**	.419	.457
Skew	x_1	1.94	**1.90±.038**	2.01±.041	**1.90±.028**	1.83	1.63
	x_2	.681	.591±.018	.628±.018	.596±.014	**.630**	.872
Kurt	x_1	8.54	7.85±.210	**8.54±.239**	8.00±.147	7.64	7.57
	x_2	3.44	3.33±.035	**3.51±.041**	3.27±.035	3.19	3.88

Table 4: Estimated posterior moments and their standard deviations for the BOD example. The closest estimates to MCMC are highlighted in bold.

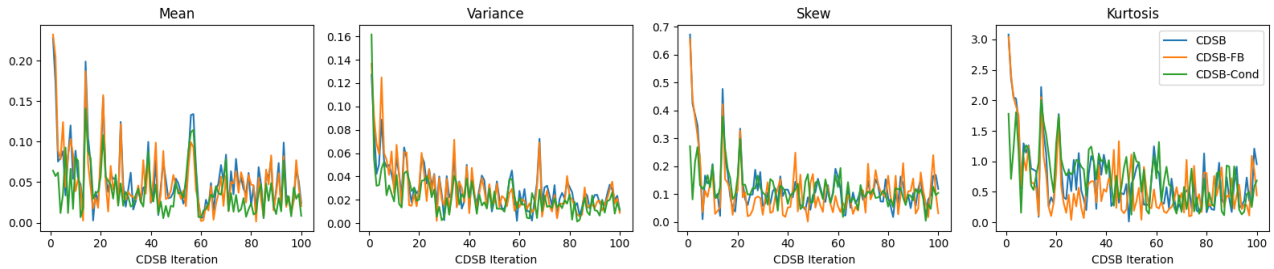


Figure 5: Convergence of estimated posterior moments with increasing number of CDSB iterations.

wal and Nichol [2021] we consider a linear schedule where $\gamma_1 = \gamma_{\text{min}}$, $\gamma_N = \gamma_{\text{max}}$, and $\gamma_k = \gamma_{\text{min}} + \frac{k-1}{N-1}(\gamma_{\text{max}} - \gamma_{\text{min}})$. In this way, the diffusion step size gets finer as the reverse process approaches $\pi_0 = p_{\text{data}}$, so as to increase the accuracy of the generated samples.

G.2 2D SYNTHETIC EXAMPLES

For the 2D examples, we use $N = 50$ diffusion steps and choose the time step schedule such that $\gamma_{\text{min}} = 10^{-4}, \gamma_{\text{max}} = 0.005$. At each IPF iteration, we train the network for 30,000 iterations using the Adam optimizer with learning rate 10^{-4} and a batch size of 100.

G.3 BIOCHEMICAL OXYGEN DEMAND MODEL

For the BOD example, we again use $N = 50$ diffusion steps with time schedule $\gamma_{\text{min}} = \gamma_{\text{max}} = 0.01$. For CDSB-Cond, we use the shortened time schedule $\gamma_{\text{min}} = \gamma_{\text{max}} = 0.005$ and a neural network regressor of the same architecture (with x and k components removed) as the conditional initialization. The batch size and optimizer settings are the same as above.

We report the estimated posterior moments as well as their standard deviation in Table 4. We further plot the convergence of RMSE for each of the statistics in Figure 5. As can be observed, IPF converges after about 20 iterations, and errors for all statistics are improved compared with CSGM (corresponding to IPF iteration 1). Using conditional initialization also helps with localizing the problem and reduces estimation errors especially in early iterations.

G.4 IMAGE EXPERIMENTS

For all image experiments, we use batch size 128 and the Adam optimizer with learning rate 10^{-4}. All methods are trained for 500k iterations. Since both $\mathbf{F}$ and $\mathbf{B}$ needs to be trained, the training time is approximately doubled for CDSB. Following Song and Ermon [2020], we make use of the exponential moving average (EMA) of the network parameters with EMA rate 0.999 at test time. We use $\gamma_{\text{min}} = 5 \times 10^{-5}$ for all experiments, and perform a parameter sweep for γ_{max} in $\{0.005, 0.01, 0.05, 0.1\}$.

G.4.1 MNIST

For the MNIST dataset, we use a U-Net architecture with 3 resolution levels each with 2 residual blocks. The numbers of filters at each resolution level are 64, 128, 128 respectively. The total number of parameters is 6.6m. Since we observe overfitting on the MNIST training set for all methods, we also apply dropout with $p = 0.1$ for the MNIST experiments. For each CDSB iteration, 100k or 250k training steps are used, corresponding to $L = 5$ or $L = 2$ CDSB iterations in total, which we find to be sufficient on this simpler dataset.

For $N = 10$, CDSB generates a minibatch of 100 images in approximately 0.8 seconds when run on a GTX 1080Ti. As a baseline comparison, we experimented with the methodology in Kadkhodaie and Simoncelli [2021] on the same MNIST test set and find that it gives PSNR/SSIM values of 15.78/0.72 and 12.49/0.47 for super-resolution and inpainting respectively (*c.f.* Table 2). Around 250 iterations are required for generating each image, or approximately 1 second generation time for 1 image on a GTX 1080Ti. In comparison, the CDSB methodology is much more efficient and achieves better image quality on both tasks.

G.4.2 CelebA

For the CelebA dataset, we use a U-Net architecture with 4 resolution levels each with 2 residual blocks and self-attention blocks at 16×16 and 8×8 resolutions. The numbers of filters at each resolution level are 128, 256, 256, 256 respectively. The total number of parameters is 39.6m. For each CDSB iteration, 10k or 25k training steps are used, corresponding to $L = 50$ or $L = 20$ CDSB iterations in total. For smaller γ_{max}, we find that higher number of CDSB iterations are beneficial.

For $N = 20, 50$, CDSB generates a minibatch of 100 images in approximately 12, 30 seconds when run on a Titan RTX. As a baseline comparison, we find that CDSB-Cond with $N = 20$ even outperforms a standard CSGM with $N = 200$, which achieves PSNR/SSIM values of approximately 20.98/0.62. To ensure that conditional initialization is not the sole contributor to the gain in sample quality, we further compare CDSB-Cond ($N = 50$) to a CSGM ($N = 50$) with conditional initialization. The forward noising process is also modified to the discretized Ornstein–Uhlenbeck process targeting $p_{\text{ref}}(x|y)$ as described in Section 4.2. This modification achieved PSNR/SSIM values of 20.84/0.59 (*c.f.* Table 2), which indicates that the CDSB framework presents larger benefits in addition to conditional initialization.

As another baseline comparison, the SNIPS algorithm [Kawar et al., 2021] reports PSNR of 21.90 for 8 CelebA test images and, when averaging across 8 predicted samples for each of the images, a PSNR of 24.31. The algorithm requires 2500 iterations for image generation, or approximately 2 minutes for producing 8 samples when run on an RTX 3080 as reported by Kawar et al. [2021]. On the same test benchmark, CDSB with $N = 50$ achieved PSNR values of 21.87 and 24.20 respectively in 3.1 seconds, thus achieving similar levels of sample quality using much less iterations. Furthermore, the SNIPS algorithm is applicable specifically for tractable linear Gaussian inverse problems, whereas CDSB is more general and does not rely on tractable likelihoods.

G.5 OPTIMAL FILTERING IN STATE-SPACE MODELS

For the sake of completeness, we first give details of the Lorenz-63 model here. It is defined using the following ODE system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x), \quad \frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y, \quad \frac{\mathrm{d}z}{\mathrm{d}t} = xy - \theta z.$$

We consider the values $\sigma = 10$, $\rho = 28$ and $\theta = 8/3$, which results in chaotic dynamics famously known as the Lorenz attractor. We integrate this system using the 4th order Runge–Kutta method with step size 0.05. For the state-space model, we define $(X_t)_{t\geq 1}$ as the states (x, y, z) of the system at regular intervals of $\delta t = 0.1$ with small Gaussian perturbations of mean 0 and variance 10^{-4}, and $(Y_t)_{t\geq 1}$ as noisy observations of $(X_t)_{t\geq 1}$ with Gaussian noise of mean 0 and variance 4. More explicitly, the transition density is thus defined as

$$f(x_t|x_{t-1}) = \mathcal{N}(x_t; \text{RK4}(x_{t-1}, 0.1), 10^{-4}), \quad g(y_t|x_t) = \mathcal{N}(y_t; x_t, 4\,\mathrm{Id}),$$

where $\text{RK4}(x_t, 0.1)$ is the 4th order Runge–Kutta operator (with step size 0.05) for the Lorenz-63 dynamics with initial condition x_t and termination time 0.1.

We run the model for 4,000 time steps and perform Bayesian filtering for the last 2,000 time steps. To accelerate the sequential inference process, we use linear regression in this example to fit $\mathbf{F}, \mathbf{B}$ with nonlinear feature expansion using

M	200	500	1000	2000
EnKF	.476±.010	.474±.005	.475±.005	.475±.003
CSGM (short)	Diverges			
CDSB (short)	.464±.013	.391±.010	.369±.007	.352±.008
CDSB-Cond (short)	**.428±.016**	**.378±.012**	**.359±.015**	**.340±.007**
CSGM (long)	**.431±.010**	.376±.008	.360±.012	.343±.006
CDSB (long)	.582±.328	.370±.012	.348±.006	.333±.006
CDSB-Cond (long)	.660±.310	**.368±.016**	**.344±.010**	**.331±.006**

(a)

M	200	500	1000	2000
EnKF	.255±.003	.286±.002	.296±.001	.300±.003
CSGM (short)	Diverges			
CDSB (short)	.203±.005	.167±.003	.150±.002	.137±.002
CDSB-Cond (short)	**.148±.004**	**.124±.002**	**.108±.002**	**.099±.001**
CSGM (long)	.204±.005	.163±.008	.140±.002	.129±.001
CDSB (long)	**.140±.008**	.129±.003	.123±.003	.120±.002
CDSB-Cond (long)	.176±.006	**.120±.002**	**.110±.003**	**.106±.002**

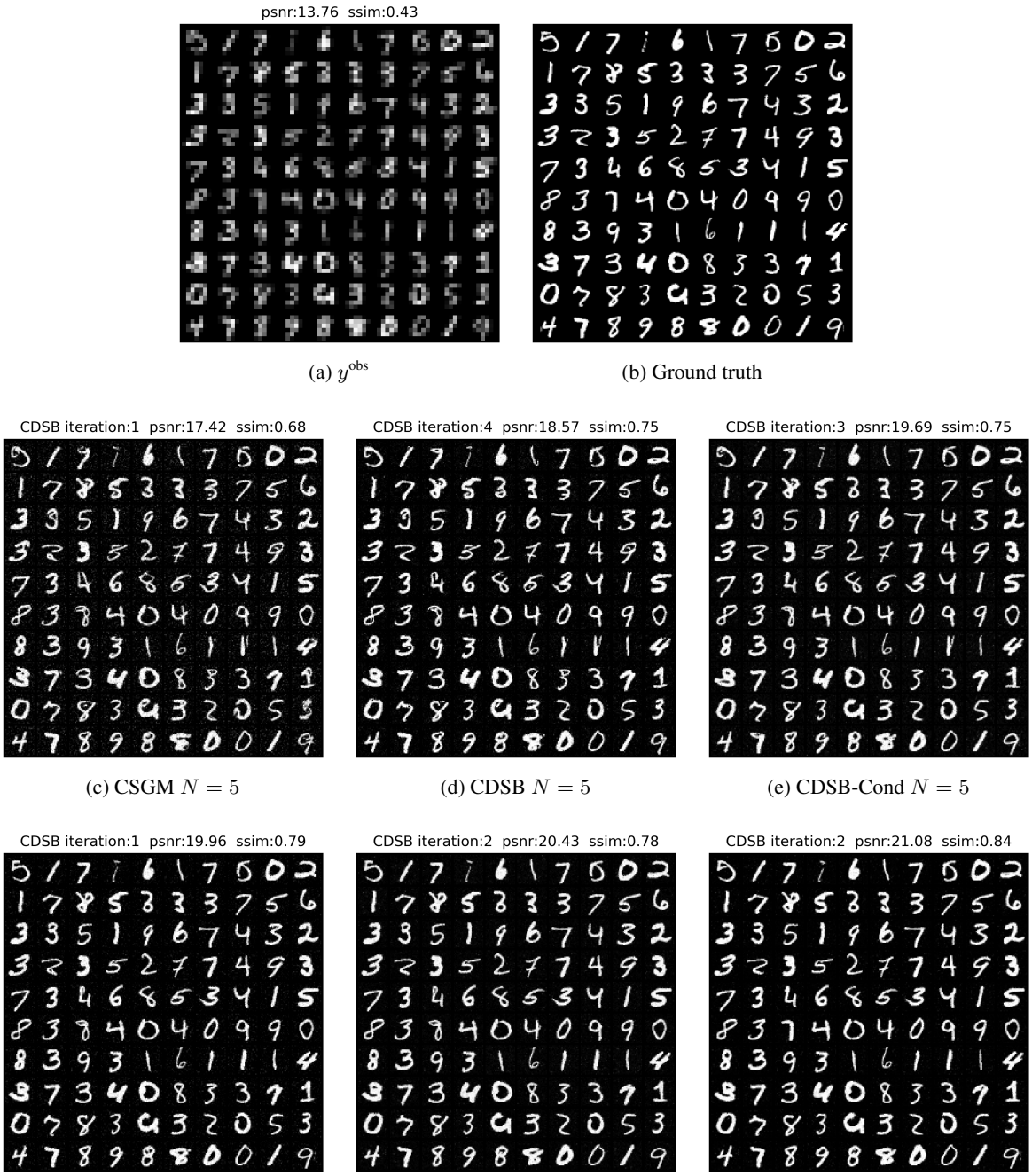
(b)

Table 5: RMSEs over 10 runs between (a) each algorithm's filtering means and the true states $x_{1:T}$ for $N = 20$ (short) and $N = 100$ (long); (b) each algorithm's filtering standard deviations and the ground truth filtering standard deviations. The lowest errors are highlighted in bold.

radial basis functions. Similar to Spantini et al. [2022], we experiment with the number of nonlinear features from 1 to 3 RBFs, in addition to the linear feature. We find that as the ensemble size M increases, increasing the number of features is helpful for lowering filtering errors, suggesting that bias-variance tradeoff is at play.

Since the system's dynamics are chaotic and can move far from the origin and display different scaling for each dimension, it is not suitable to choose $p_{\text{ref}}(x) = \mathcal{N}(x; 0, \text{Id})$. Therefore, for CSGM and CDSB, we let $p_{\text{ref}}(x) = \mathcal{N}(x; \mu_{\text{ref}}, \text{diag}(\sigma^2_{\text{ref}}))$ where $\mu_{\text{ref}}, \sigma^2_{\text{ref}}$ are the estimated mean and variance of the prior predictive distribution $p(x_t | y^{\text{obs}}_{1:t-1})$ at time t. For CDSB-Cond, we let $p_{\text{ref}}(x|y) = \mathcal{N}(x; \mu_{\text{ref}}, \text{diag}(\sigma^2_{\text{ref}}))$ where the estimated posterior mean and variance are returned by EnKF. Furthermore, we scale the diffusion process's time step dimensionwise by the variance of the reference measure σ^2_{ref}. We consider a short diffusion process with $N = 20$, and a long diffusion process with $N = 100$. We let $\gamma_{\text{min}} = 0.0005 \cdot \sigma^2_{\text{ref}}$ and $\gamma_{\text{max}} = 0.05 \cdot \sigma^2_{\text{ref}}$ for the short diffusion process, and reduce γ_{max} by a half for the long diffusion process.

We report the RMSEs between each algorithm's filtering means and the ground truth filtering means in Table 3. We compute the ground truth filtering means using a particle filter with $M = 10^6$ particles. In addition, we report the RMSEs between each algorithm's filtering means and the true states $x_{1:T}$ in Table 5a, and between each algorithm's filtering standard deviations and the ground truth standard deviations in Table 5b. Similarly, we observe that CDSB and CDSB-Cond achieve lower errors than CSGM and EnKF. In the case where the ensemble size $M = 200$, however, when using the long diffusion process we observe occasional large errors for CDSB and CDSB-Cond. We conjecture that since CDSB is an iterative algorithm, inevitably small errors in regression can be accumulated. For small ensemble size and large number of diffusion steps, the model may thus be more prone to overfitting. However, for larger ensemble size $M \geq 500$, we do not observe this issue, and CDSB-Cond achieves the lowest RMSE.

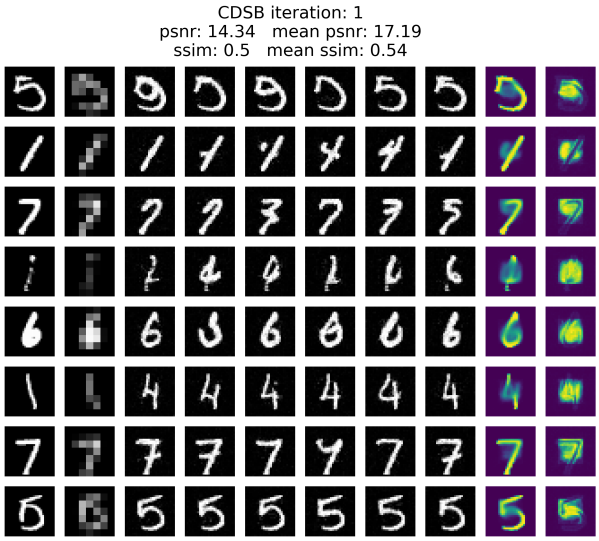


(a) y^{obs} (b) Ground truth

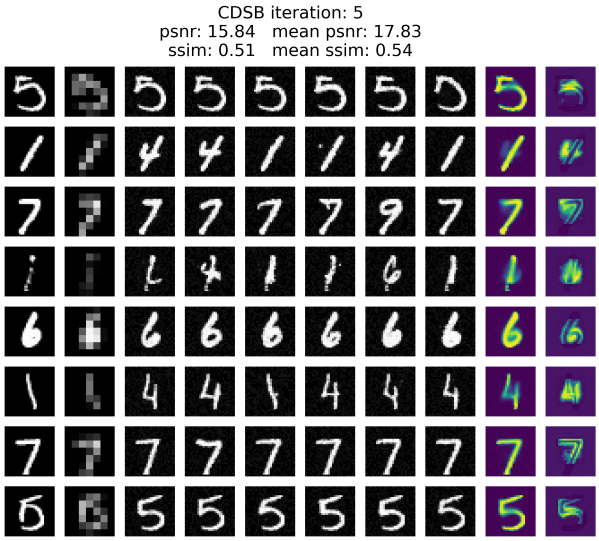
(c) CSGM $N = 5$ (d) CDSB $N = 5$ (e) CDSB-Cond $N = 5$

(f) CSGM $N = 10$ (g) CDSB $N = 10$ (h) CDSB-Cond $N = 10$

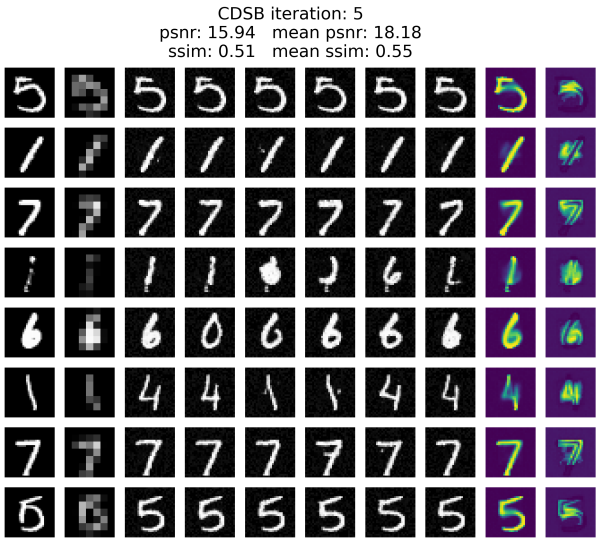
Figure 6: Additional samples for the MNIST 4x super-resolution task.



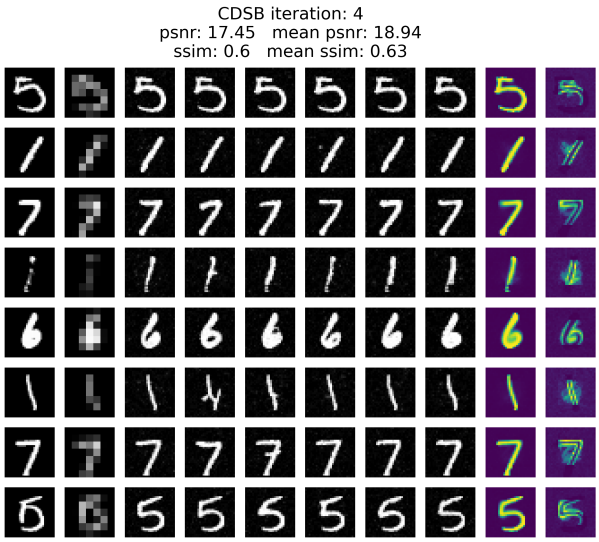
(a) CSGM $N = 10$



(b) CDSB $N = 10$



(c) CDSB-FB $N = 10$

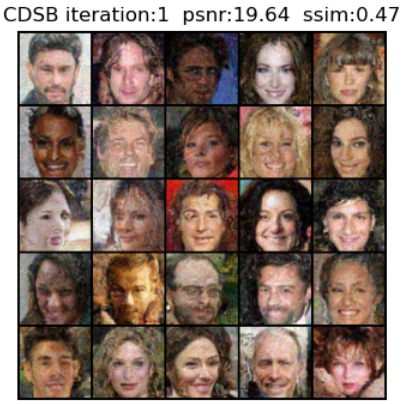


(d) CDSB-Cond $N = 10$

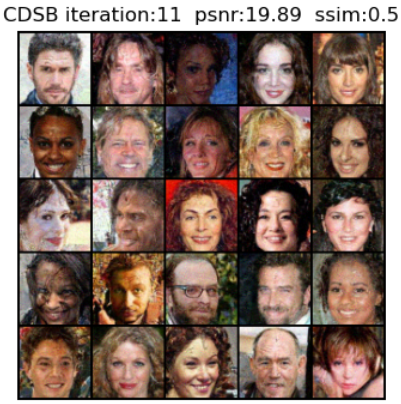
Figure 7: Uncurated conditional samples for the MNIST 14x14 inpainting task. The first two columns correspond to ground truth, y^{obs}, and the last two columns correspond to the mean and standard deviation of 100 samples.

(a) y^{obs}

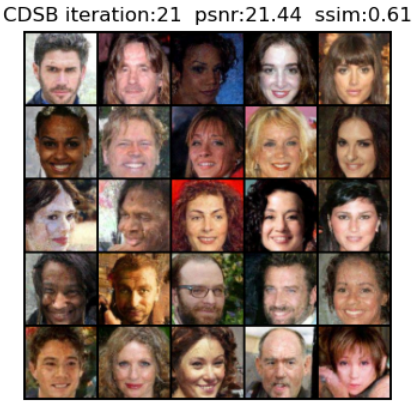
(b) Ground truth



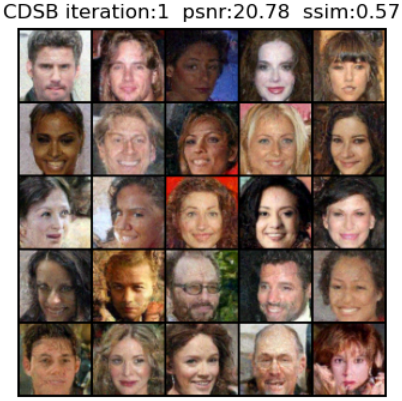
(c) CSGM $N = 20$



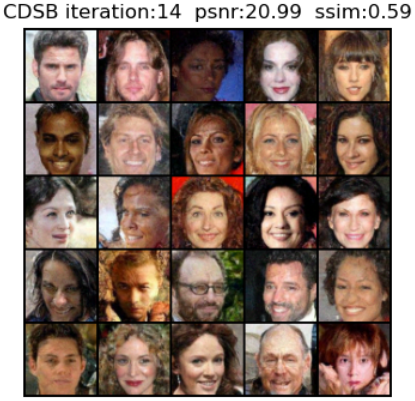
(d) CDSB $N = 20$



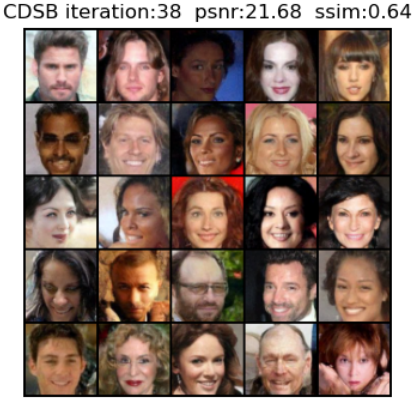
(e) CDSB-Cond $N = 20$



(f) CSGM $N = 50$

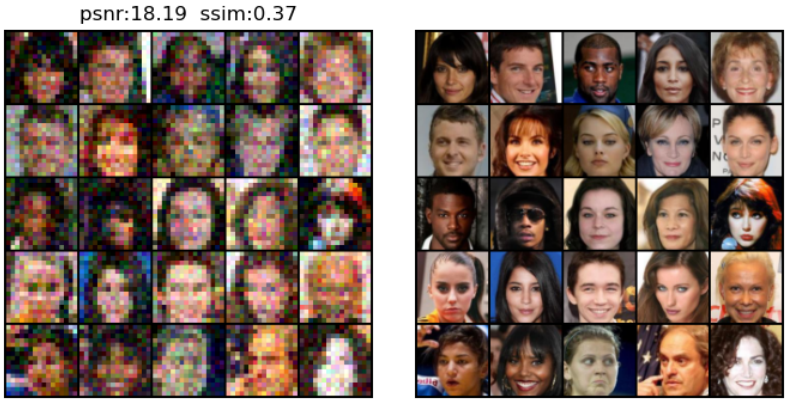


(g) CDSB $N = 50$

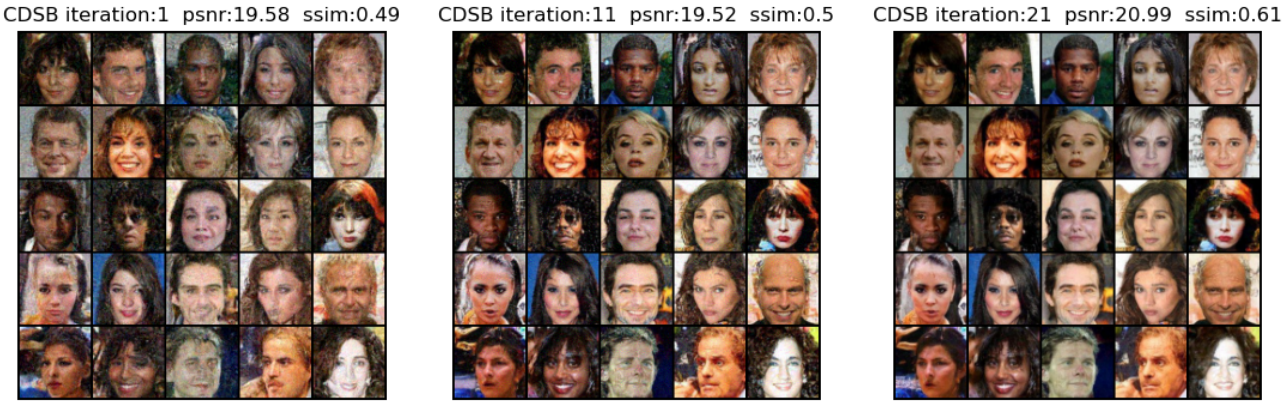


(h) CDSB-Cond $N = 50$

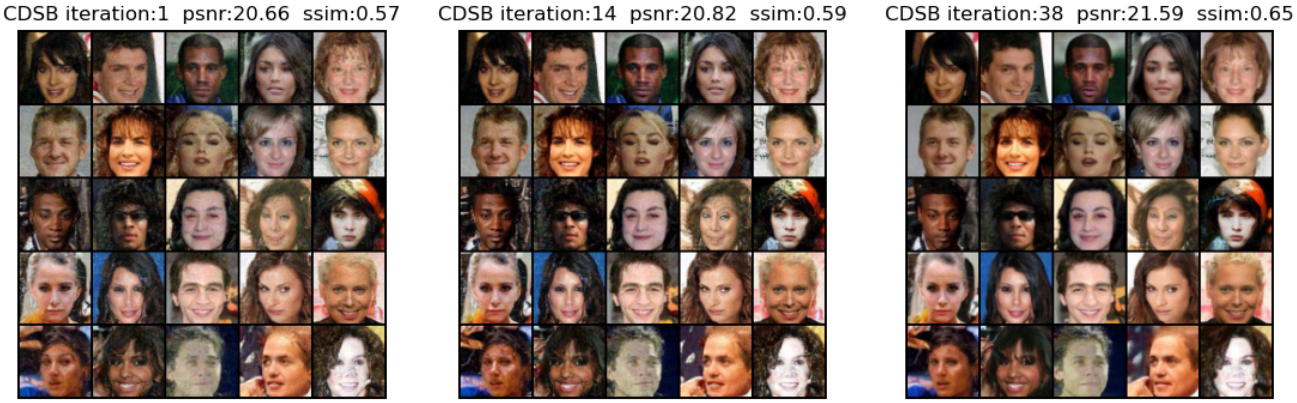
Figure 8: Uncurated samples for the CelebA 4x super-resolution with Gaussian noise task.



(a) y^{obs} (b) Ground truth



(c) CSGM $N = 20$ (d) CDSB $N = 20$ (e) CDSB-Cond $N = 20$



(f) CSGM $N = 50$ (g) CDSB $N = 50$ (h) CDSB-Cond $N = 50$

Figure 9: Uncurated samples for the CelebA 4x super-resolution with Gaussian noise task.

Figure 10: Uncurated conditional samples using CDSB-Cond with $N = 50$ for the CelebA 4x super-resolution with Gaussian noise task. The first two columns correspond to ground truth, y^{obs}, and the last column corresponds to the mean of the middle 8 samples.