

---

# Monte Carlo guided Diffusion for Bayesian linear inverse problems

---

**Gabriel Cardoso**

CMAP, Ecole polytechnique,  
IHU Liryc,  
Fondation Bordeaux Université

**Yazid Janati El Idrissi**

CMAP, Ecole polytechnique,  
CITI, Télécom SudParis

**Sylvain Le Corff**

LPSM, Sorbonne Université

**Eric Moulines**

CMAP, Ecole polytechnique

## Abstract

Ill-posed linear inverse problems arise frequently in various applications, from computational photography to medical imaging. A recent line of research exploits Bayesian inference with informative priors to handle the ill-posedness of such problems. Amongst such priors, score-based generative models (SGM) have recently been successfully applied to several different inverse problems. In this study, we exploit the particular structure of the prior defined by the SGM to define a sequence of intermediate linear inverse problems. As the noise level decreases, the posteriors of these inverse problems get closer to the target posterior of the original inverse problem. To sample from this sequence of posteriors, we propose the use of Sequential Monte Carlo (SMC) methods. The proposed algorithm, MCGdiff, is shown to be theoretically grounded and we provide numerical simulations showing that it outperforms competing baselines when dealing with ill-posed inverse problems in a Bayesian setting.

## 1 Introduction

This paper is concerned with linear inverse problems  $y = Ax + \sigma_y \varepsilon$ , where  $y \in \mathbb{R}_y^d$  is a vector of indirect observations,  $x \in \mathbb{R}^{d_x}$  is the vector of unknowns,  $A \in \mathbb{R}^{d_y \times d_x}$  is the linear forward operator and  $\varepsilon \in \mathbb{R}^{d_y}$  is an unknown noise vector. This general model is used throughout computational imaging, including various tomographic imaging applications such as common types of magnetic resonance imaging [Vlaardingerbroek and Boer, 2013], X-ray computed tomography [Elbakri and Fessler, 2002], radar imaging [Cheney and Borden, 2009], and basic image restoration tasks such as deblurring, superresolution, and image inpainting [González et al., 2009]. The classical approach to solving linear inverse problems relies on prior knowledge about  $x$ , such as its smoothness, sparseness in a dictionary, or its geometric properties. These approaches attempt to estimate a  $\hat{x}$  by minimizing a regularized inverse problem,  $\hat{x} = \operatorname{argmin}_x \{\|y - Ax\|^2 + \text{Reg}(x)\}$ , where  $\text{Reg}$  is a regularization term that balances data fidelity and noise while enabling efficient computations. However, a common difficulty in the regularized inverse problem is the selection of an appropriate regularizer, which has a decisive influence on the quality of the reconstruction.

Whereas regularized inverse problems continue to dominate the field, many alternative statistical formulations have been proposed; see [Besag et al., 1991, Idier, 2013, Marnissi et al., 2017] and the references therein - see [Stuart, 2010] for a mathematical perspective. A main advantage of statistical approaches is that they allow for **uncertainty quantification** in the reconstructed solution; see [Dashti and Stuart, 2017]. The **Bayes' formulation** of the regularized inverse problem is based on considering the indirect measurement  $Y$ , the state  $X$  and the

---

Corresponding authors: gabriel.victorino-cardoso@polytechnique.edu,  
yazid.janati@polytechnique.edu

noise  $\varepsilon$  as random variables, and to specify  $p(y|x)$  the *likelihood* (the conditional distribution of  $Y$  at  $X$ ) and the prior  $p(x)$  (the distribution of the state). One can use Bayes' theorem to obtain the **posterior distribution**  $p(x|y) \propto p(y|x)p(x)$ , where " $\propto$ " means that the two sides are equal to each other up to a multiplicative constant that does not depend on  $x$ . Moreover, the use of an appropriate method for Bayesian inference allows the quantification of the uncertainty in the reconstructed solution  $x$ . A variety of priors are available, including but not limited to Laplace [Figueiredo et al., 2007], total variation (TV) [Kaipio et al., 2000] and mixture-of-Gaussians [Fergus et al., 2006]. In the last decade, a variety of techniques have been proposed to design and train generative models capable of producing perceptually realistic samples from the original data, even in challenging high-dimensional data such as images or audio [Kingma et al., 2019, Kobyzhev et al., 2020, Gui et al., 2021]. Denoising diffusion models have been shown to be particularly effective generative models in this context [Sohl-Dickstein et al., 2015, Song et al., 2021c, Song et al., 2021a, Song et al., 2021b, Benton et al., 2022]. These models convert noise into the original data domain through a series of denoising steps. A popular approach is to use a generic diffusion model that has been pre-trained, eliminating the need for re-training and making the process more efficient and versatile [Trippe et al., 2023, Zhang et al., 2023]. Although this was not the main motivation for developing these models, they can of course be used as prior distributions in Bayesian inverse problems. This simple observation has led to a new, fast-growing line of research on how linear inverse problems can benefit from the flexibility and expressive power of the recently introduced deep generative models; see [Arjomand Bigdeli et al., 2017, Wei et al., 2022, Su et al., 2022, Kaltenbach et al., 2023, Shin and Choi, 2023, Zhihang et al., 2023, Sahlström and Tarvainen, 2023].

## Contributions

- We propose MCGdiff, a novel algorithm for sampling from the Bayesian posterior of Gaussian linear inverse problems with denoising diffusion model priors. MCGdiff specifically exploits the structure of both the linear inverse problem and the denoising diffusion generative model to design an efficient SMC sampler.
- We establish under sensible assumptions that the empirical distribution of the samples produced by MCGdiff converges to the target posterior when the number of particles goes to infinity. To the best of our knowledge, MCGdiff is the first provably consistent algorithm for conditional sampling from the denoising diffusion posteriors.
- To evaluate the performance of MCGdiff, we perform numerical simulations on several examples for which the target posterior distribution is known. Simulation results support our theoretical results, i.e. the empirical distribution of samples from MCGdiff converges to the target posterior distribution. This is **not** the case for the competing methods (using the same denoising diffusion generative priors) which are shown, when run with random initialization of the denoising diffusion, to generate a significant number of samples outside the support of the target posterior. We also illustrate samples from MCGdiff in imaging inverse problems.

**Background and notations.** This section provides a concise overview of the diffusion model framework and notations used in this paper. We cover the elements that are important for understanding our approach, and we recommend that readers refer to the original papers for complete details and derivations [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021c, Song et al., 2021a]. A denoising diffusion model is a generative model consisting of a forward and a backward process. The forward process involves sampling  $X_0 \sim q_{\text{data}}$  from the data distribution, which is then converted to a sequence  $X_{1:n}$  of recursively corrupted versions of  $X_0$ . The backward process involves sampling  $X_n$  according to an easy-to-sample reference distribution on  $\mathbb{R}^{d_x}$  and generating  $X_0 \in \mathbb{R}^{d_x}$  by a sequence of denoising steps. Following [Sohl-Dickstein et al., 2015, Song et al., 2021a], the forward process can be chosen as a Markov chain with joint distribution

$$q_{0:n}(x_{0:n}) = q_{\text{data}}(x_0) \prod_{t=1}^n q_t(x_t|x_{t-1}), \quad q_t(x_t|x_{t-1}) = \mathcal{N}(x_t; (1 - \beta_t)^{1/2} x_{t-1}, \beta_t I_{d_x}), \quad (1.1)$$

where  $I_{d_x}$  is the identity matrix of size  $d_x$ ,  $\{\beta_t\}_{t \in \mathbb{N}} \subset (0, 1)$  is a non-increasing sequence and  $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$  is the p.d.f. of the Gaussian distribution with mean  $\mu$  and covariance matrix  $\Sigma$  (assumed to be non-singular) evaluated at  $\mathbf{x}$ . For all  $t > 0$ , set  $\bar{\alpha}_t = \prod_{\ell=1}^t (1 - \beta_\ell)$  with the convention  $\alpha_0 = 1$ . We have for all  $0 \leq s < t \leq n$ ,

$$q_{t|s}(x_t|x_s) := \int \prod_{\ell=s+1}^t q_\ell(x_\ell|x_{\ell-1}) dx_{s+1:t-1} = \mathcal{N}(x_t; (\bar{\alpha}_t/\bar{\alpha}_s)^{1/2} x_s, (1 - \bar{\alpha}_t/\bar{\alpha}_s) I_{d_x}). \quad (1.2)$$

For the standard choices of  $\bar{\alpha}_t$ , the sequence of distributions  $(q_t)_t$  converges weakly to the standard normal distribution as  $t \rightarrow \infty$ , which we chose as the reference distribution. For the reverse process, [Song et al., 2021a, Song et al., 2021b] introduce an *inference distribution*  $q_{1:n|0}^\sigma(x_{1:n}|x_0)$ ,

depending on a sequence  $\{\sigma_t\}_{t \in \mathbb{N}}$  of hyperparameters satisfying  $\sigma_t^2 \in [0, 1 - \bar{\alpha}_{t-1}]$  for all  $t \in \mathbb{N}^*$ , and defined as  $q_{1:n|0}^\sigma(x_{1:n}|x_0) = q_{n|0}^\sigma(x_n|x_0) \prod_{t=n}^2 q_{t-1|t,0}^\sigma(x_{t-1}|x_t, x_0)$ , where  $q_{n|0}^\sigma(x_n|x_0) = \mathcal{N}(x_n; \bar{\alpha}_n^{1/2}x_0, (1 - \bar{\alpha}_n)\mathbf{I})$  and  $q_{t-1|t,0}^\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_t(x_0, x_t), \sigma_t^2 \mathbf{I}_d)$ , with  $\boldsymbol{\mu}_t(x_0, x_t) = \bar{\alpha}_{t-1}^{1/2}x_0 + (1 - \bar{\alpha}_{t-1} - \sigma_t^2)^{1/2}(x_t - \bar{\alpha}_t^{1/2}x_0)/(1 - \bar{\alpha}_t)^{1/2}$ . For  $t = n-1, \dots, 1$ , we define by backward induction the sequence  $q_{t|0}^\sigma(x_t|x_0) = \int q_{t|t+1,0}^\sigma(x_t|x_{t+1}, x_0) q_{t+1|0}^\sigma(x_{t+1}|x_0) dx_{t+1}$ . It is shown in [Song et al., 2021a, Lemma 1] that for all  $t \in [1 : n]$ , the distributions of the forward and inference process conditioned on the initial state coincide, i.e. that  $q_{t|0}^\sigma(x_t|x_0) = q_{t|0}(x_t|x_0)$ . The backward process is derived from the inference distribution by replacing, for each  $t \in [2 : n]$ ,  $x_0$  in the definition  $q_{t-1|t,0}^\sigma(x_{t-1}|x_t, x_0)$  with a prediction where  $\chi_{0|t}^\theta(x_t) := \bar{\alpha}_t^{-1/2} (x_t - (1 - \bar{\alpha}_t)^{1/2} \mathbf{e}^\theta(x_t, t))$  where  $\mathbf{e}^\theta(x, t)$  is typically a neural network parameterized by  $\theta$ . More formally, the backward distribution is defined as  $p_{0:n}^\theta(x_{0:n}) = p_n(x_n) \prod_{t=0}^{n-1} p_t^\theta(x_t|x_{t+1})$ , where  $p_n(x_n) = \mathcal{N}(x_n; \mathbf{0}_{\mathbf{d}_x}, \mathbf{I}_{\mathbf{d}_x})$  and for all  $t \in [1 : n-1]$ ,

$$p_t^\theta(x_t|x_{t+1}) := q_{t|t+1,0}^\sigma(x_t|x_{t+1}, \chi_{0|t+1}^\theta(x_{t+1})) = \mathcal{N}(x_t, \mathbf{m}_{t+1}^\theta(x_{t+1}), \sigma_{t+1}^2 \mathbf{I}_{\mathbf{d}_x}), \quad (1.3)$$

where  $\mathbf{m}_{t+1}(x_{t+1}) := \boldsymbol{\mu}(\chi_{0|t+1}^\theta(x_{t+1}), x_{t+1})$  and  $\mathbf{0}_{\mathbf{d}_x}$  is the null vector of size  $\mathbf{d}_x$ . At step 0, we set  $p_0(x_0|x_1) := \mathcal{N}(x_0; \chi_{0|1}^\theta(x_1), \sigma_1^2 \mathbf{I}_{\mathbf{d}_x})$ . The parameter  $\theta$  is obtained [Song et al., 2021a, Theorem 1] by solving the following optimization problem:

$$\theta_* \in \operatorname{argmin}_\theta \sum_{t=1}^n (2\mathbf{d}_x \sigma_t^2 \alpha_t)^{-1} \int \|\epsilon - \mathbf{e}^\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|_2^2 \mathcal{N}(\epsilon; \mathbf{0}_{\mathbf{d}_x}, \mathbf{I}_{\mathbf{d}_x}) \mathbf{q}_{\text{data}}(dx_0) d\epsilon. \quad (1.4)$$

Thus,  $\mathbf{e}^{\theta_*}(X_t, t)$  might be seen as the predictor of the noise added to  $X_0$  to obtain  $X_t$  (in the forward pass) and justifies the "prediction" terminology. The time 0 marginal  $p_0^{\theta_*}(x_0) = \int p_{0:n}^{\theta_*}(x_{0:n}) dx_{1:n}$  which we will refer to as the *prior* is used as an approximation of  $\mathbf{q}_{\text{data}}$  and the time  $s$  marginal is  $p_s^{\theta_*}(x_s) = \int p_{0:n}^{\theta_*}(x_{0:n}) dx_{1:s-1} dx_{s+1:n}$ . In the rest of the paper we drop the dependence on the parameter  $\theta_*$ . We define for all  $v \in \mathbb{R}^\ell, w \in \mathbb{R}^k$ , the concatenation operator  $v \frown w = [v^T, w^T]^T \in \mathbb{R}^{\ell+k}$ . For  $i \in [1 : \ell]$ , we let  $v[i]$  the  $i$ -th coordinate of  $v$ .

**Related works.** The subject of Bayesian problems is very vast, and it is impossible to discuss here all the results obtained in this very rich literature. One of such domains is image restoration problems, such as deblurring, denoising inpainting, which are challenging problems in computer vision that involves restoring a partially observed degraded image. Deep learning techniques are widely used for this task [Arjomand Bigdeli et al., 2017, Yeh et al., 2018, Xiang et al., 2023, Wei et al., 2022] with many of them relying on auto-encoders, VAEs [Ivanov et al., 2018, Peng et al., 2021, Zheng et al., 2019], GANs [Yeh et al., 2018, Zeng et al., 2022], or autoregressive transformers [Yu et al., 2018, Wan et al., 2021]. In what follows, we focus on methods based on denoising diffusion that has recently emerged as a way to produce high-quality realistic samples from the original data distribution on par with the best GANs in terms of image and audio generation, without the intricacies of adversarial training; see [Sohl-Dickstein et al., 2015, Song et al., 2021c, Song et al., 2022]. Diffusion-based approaches do not require specific training for degradation types, making them much more versatile and computationally efficient. In [Song et al., 2022], noisy linear inverse problems are proposed to be solved by diffusing the degraded observation forward, leading to intermediate observations  $\{y_s\}_{s=0}^n$ , and then running a modified backward process that promotes consistency with  $y_s$  at each step  $s$ . The Denoising-Diffusion-Restoration model (DDRM) [Kawar et al., ] also modifies the backward process so that the unobserved part of the state follows the backward process while the observed part is obtained as a noisy weighted sum between the noisy observation and the prediction of the state. As observed by [Lugmayr et al., 2022], DDRM is very efficient, but the simple blending used occasionally causes inconsistency in the restoration process. DPS [Chung et al., 2023] considers a backward process targeting the posterior. DPS approximates the score of the posterior using the Tweedie formula, which incorporates the learned score of the prior. The approximation error is quantified and shown to decrease when the noise level is large, i.e., when the posterior is close to the prior distribution. As shown in Section 3 with a very simple example, neither DDRM nor DPS can be used to sample the target posterior and therefore do not solve the Bayesian recovery problem (even if we run DDRM and DPS several times with independent initializations). Indeed, we show that DDRM and DPS produce samples under the "prior" distribution (which is generally captured very well by the denoising diffusion model), but which are not consistent with the observations (many samples land in areas with very low likelihood). In [Trippe et al., 2023] the authors introduce SMCdiff, a Sequential Monte Carlo-based denoising diffusion model that aims at solving specifically the *inpainting problem*. SMCdiff produces a particle approximation of the conditional distribution of the non

observed part of the state conditionally on a forward-diffused trajectory of the observation. The resulting particle approximation is shown to converge to the true posterior of the SGM under the assumption that the joint laws of the forward and backward processes coincide, which fails to be true in realistic setting. In comparison with SMCdiff, MCGdiff is a versatile approach that solves any Bayesian linear inverse problem while being consistent under practically no assumption. In parallel to our work, [Wu et al., 2023] also developed a similar SMC based methodology but with a different proposal kernel.

## 2 The MCGdiff algorithm

In this section we present our methodology for the inpainting problem (2.1), both with noise and without noise. The more general case is treated in Section 2.1. Let  $d_y \in [1 : d_x - 1]$ . In what follows we denote the  $d_y$  top coordinates of a vector  $x \in \mathbb{R}^{d_x}$  by  $\bar{x}$  and the remaining coordinates by  $\underline{x}$ , so that  $x = \bar{x}^\top \underline{x}$ . The inpainting problem is defined as

$$Y = \bar{X} + \sigma_y \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_{d_y}), \quad \sigma \geq 0, \quad (2.1)$$

where  $\bar{X}$  are the first  $d_y$  coordinates of a random variable  $X \sim p_0$ . The goal is then to recover the law of the complete state  $X$  given a realisation  $y$  of the incomplete observation  $\mathbf{Y}$  and the model (2.1).

**Noiseless case.** We begin by the case  $\sigma_y = 0$ . As the first  $d_y$  coordinates are observed exactly, we aim at inferring the remaining coordinates of  $X$ , which correspond to  $\underline{X}$ . As such, given an observation  $y$ , we aim at sampling from the posterior  $\phi_0^y(\underline{x}_0) \propto p_0(y^\top \underline{x}_0 | x_1)$  with integral form

$$\phi_0^y(\underline{x}_0) \propto \int p_n(x_n) \left\{ \prod_{s=1}^{n-1} p_s(x_s | x_{s+1}) \right\} p_0(y^\top \underline{x}_0 | x_1) dx_{1:n}. \quad (2.2)$$

To solve this problem, we propose to use SMC algorithms [Doucet et al., 2001, Cappé et al., 2005, Chopin et al., 2020], where a set of  $N$  random samples, referred to as particles, is iteratively updated to approximate the posterior distribution. The updates involve, at iteration  $s$ , selecting promising particles from the pool of particles  $\xi_{s+1}^{1:N} = (\xi_{s+1}^1, \dots, \xi_{s+1}^N)$  based on a weight function  $\tilde{\omega}_s$ , and then apply a Markov transition  $p_s^y$  to obtain the samples  $\xi_s^{1:N}$ . The transition  $p_s^y(x_s | x_{s+1})$  is designed to follow the backward process while guiding the  $d_y$  top coordinates of the pool of particles  $\xi_s^{1:N}$  towards the measurement  $y$ . Note that under the backward dynamics (1.3),  $\bar{X}_t$  and  $\underline{X}_t$  are independent conditionally on  $X_{t+1}$  with transition kernels respectively  $\bar{p}_t(\bar{x}_t | x_{t+1}) := \mathcal{N}(\bar{x}_t; \bar{m}_{t+1}(x_{t+1}), \sigma_{t+1}^2 I_{d_y})$  and  $\underline{p}_t(\underline{x}_t | x_{t+1}) := \mathcal{N}(\underline{x}_t; \underline{m}_{t+1}(x_{t+1}), \sigma_{t+1}^2 I_{d_x - d_y})$  where  $\bar{m}_{t+1}(x_{t+1}) \in \mathbb{R}^{d_y}$  and  $\underline{m}_{t+1}(x_{t+1}) \in \mathbb{R}^{d_x - d_y}$  are such that  $\underline{m}_{t+1}(x_{t+1}) = \bar{m}_{t+1}(x_{t+1})^\top \underline{m}_{t+1}(x_{t+1})$  and the above kernels satisfy  $p_t(x_s | x_{s+1}) = \bar{p}_t(\bar{x}_t | x_{t+1}) \underline{p}_t(\underline{x}_t | x_{t+1})$ . We consider the following proposal kernels for  $t \in [1 : n]$ ,

$$p_s^y(x_t | x_{t+1}) \propto p_t(x_t | x_{t+1}) \bar{q}_{t|0}(\bar{x}_t | y), \quad \text{where } \bar{q}_{t|0}(\bar{x}_t | y) := \mathcal{N}(\bar{x}_t; \bar{\alpha}_t^{1/2} y, (1 - \bar{\alpha}_t) I_{d_y}), \quad (2.3)$$

and  $p_n^y(x_n) \propto p_n(x_n) \bar{q}_{n|0}(\bar{x}_n | y)$ . For the final step, we define  $\underline{p}_0^y(\underline{x}_0 | x_1) = \underline{p}_0(\underline{x}_0 | x_1)$ . Using standard Gaussian conjugation formulas, we obtain

$$\begin{aligned} p_t^y(x_t | x_{t+1}) &= \underline{p}_t(\underline{x}_t | x_{t+1}) \cdot \mathcal{N}(\bar{x}_t; \mathsf{K}_t \bar{\alpha}_t^{1/2} y + (1 - \mathsf{K}_t) \bar{m}_{t+1}(x_{t+1}), (1 - \bar{\alpha}_t) \mathsf{K}_t \cdot I_{d_y}), \\ p_n(x_n) &= \mathcal{N}(\underline{x}_n; 0_{d_x - d_y}, I_{d_x - d_y}) \cdot \mathcal{N}(\bar{x}_n; \mathsf{K}_n \bar{\alpha}_n^{1/2} y, (1 - \bar{\alpha}_n) \mathsf{K}_n \cdot I_{d_y}) \end{aligned}$$

where  $\mathsf{K}_t := \sigma_{t+1}^2 / (\sigma_{t+1}^2 + 1 - \alpha_t)$ . For this procedure to target the posterior  $\phi_0^y$ , the weight function  $\tilde{\omega}_s$  is chosen as follows; we set  $\tilde{\omega}_{n-1}(x_n) := \int p_{n-1}(x_{n-1} | x_n) \bar{q}_{n-1|0}(\bar{x}_{n-1} | y) dx_{n-1} = \mathcal{N}(\bar{\alpha}_{n-1}^{1/2} y; \bar{m}_n(x_n), \sigma_n^2 + 1 - \alpha_s)$  and for  $t \in [1 : n - 2]$ ,

$$\tilde{\omega}_t(x_{t+1}) := \frac{\int \bar{p}_t(\bar{x}_t | x_{t+1}) \bar{q}_{t|0}(\bar{x}_t | y) dx_t}{\bar{q}_{t+1|0}(\bar{x}_{t+1} | y)} = \frac{\mathcal{N}(\bar{\alpha}_t^{1/2} y; \bar{m}_{t+1}(x_{t+1}), (\sigma_{t+1}^2 + 1 - \alpha_t) I_{d_y})}{\mathcal{N}(\bar{\alpha}_{t+1}^{1/2} y; \bar{x}_{t+1}, (1 - \alpha_{t+1}) I_{d_y})}. \quad (2.4)$$

For the final step, we set  $\tilde{\omega}_0(x_1) := \bar{p}_0(y | \bar{x}_1) / \bar{q}_{1|0}(\bar{x}_1 | y)$ . The overall SMC algorithm targeting  $\phi_0^y$  using the instrumental kernel (2.3) and weight function (2.4) is summarized in Algorithm 1. We now provide a justification to Algorithm 1. Let  $\{g_s^y\}_{s=1}^n$  be a sequence of positive functions. Consider the sequence of distributions  $\{\phi_s^y\}_{s=1}^n$

---

**Algorithm 1:** MCGdiff ( $\sigma = 0$ )

---

**Input:** Number of particles  $N$   
**Output:**  $\xi_0^{1:N}$

// Operations involving index  $i$  are repeated for each  $i \in [1 : N]$

$$\bar{z}_n^i \sim \mathcal{N}(\mathbf{0}_{d_y}, \mathbf{I}_{d_y}), \quad \underline{z}_n^i \sim \mathcal{N}(\mathbf{0}_{d_x-d_y}, \mathbf{I}_{d_x-d_y}), \quad \bar{\xi}_n^i = \mathsf{K}_n \bar{\alpha}_n^{1/2} y + (1 - \bar{\alpha}_n) \mathsf{K}_n \bar{z}_n^i, \quad \xi_n^i = \bar{\xi}_n^i \cap \underline{z}_n^i;$$

**for**  $s \leftarrow n - 1 : 0$  **do**

**if**  $s = n - 1$  **then**

$\tilde{\omega}_{n-1}(\xi_n^i) = \mathcal{N}(\bar{\alpha}_n^{1/2} y; \bar{\mathbf{m}}_n(\xi_n^i), 2 - \bar{\alpha}_n);$

**else**

$\tilde{\omega}_s(\xi_{s+1}^i) = \mathcal{N}(\bar{\alpha}_s^{1/2} y; \bar{\mathbf{m}}_{s+1}(\xi_{s+1}^i), \sigma_{s+1}^2 + 1 - \bar{\alpha}_s) / \mathcal{N}(\bar{\alpha}_{s+1}^{1/2} y; \bar{\xi}_{s+1}^i, 1 - \bar{\alpha}_{s+1});$

$I_{s+1}^i \sim \text{Cat}(\{\tilde{\omega}_s(\xi_{s+1}^j) / \sum_{k=1}^N \tilde{\omega}_s(\xi_{s+1}^k)\}_{j=1}^N), \quad \bar{z}_s^i \sim \mathcal{N}(\mathbf{0}_{d_y}, \mathbf{I}_{d_y}), \quad \underline{z}_s^i \sim \mathcal{N}(\mathbf{0}_{d_x-d_y}, \mathbf{I}_{d_x-d_y});$

$\bar{\xi}_s^i = \mathsf{K}_s \bar{\alpha}_s^{1/2} y + (1 - \mathsf{K}_s) \bar{\mathbf{m}}_{s+1}(\xi_{s+1}^i) + (1 - \alpha_s)^{1/2} \mathsf{K}_s^{1/2} \bar{z}_s^i, \quad \xi_s^i = \underline{\mathbf{m}}_{s+1}(\xi_{s+1}^i) + \sigma_{s+1} \underline{z}_s^i;$

Set  $\xi_s^i = \bar{\xi}_s^i \cap \xi_s^i;$

---

defined as follows;  $\phi_n^y(x_n) \propto p_n(x_n) g_n^y(x_n)$  and for  $t \in [1 : n - 1]$

$$\phi_t^y(x_t) \propto \int g_{t+1}^y(x_{t+1})^{-1} g_t^y(x_t) p_t(x_t | x_{t+1}) \phi_{t+1}^y(dx_{t+1}), \quad (2.5)$$

By construction, the time  $t$  marginal (2.5) is  $\phi_t^y(x_t) \propto p_t(x_t) g_t^y(x_t)$  for all  $t \in [1 : n]$ . Then, using  $\phi_1^y$  and (2.2), we have that  $\phi_0^y(\underline{x}_0) \propto \int g_1^y(x_1)^{-1} \bar{p}_0(y | \bar{x}_1) p_0(\underline{x}_0 | x_1) \phi_1^y(dx_1)$ .

The recursion (2.5) suggests a way of obtaining a particle approximation of  $\phi_0^y$ ; by sequentially approximating each  $\phi_t^y$  we can effectively derive a particle approximation of the posterior. To construct the intermediate particle approximations we use the framework of *auxiliary particle filters* (APF) [Pitt and Shephard, 1999]. We focus on the case  $g_t^y(x_t) = \bar{q}_{t|0}(\bar{x}_t | y)$  which corresponds to Algorithm 1. The initial particle approximation  $\phi_n^y$  is obtained by drawing  $N$  i.i.d. samples  $\xi_n^{1:N}$  from  $p_n^y$  and setting  $\phi_n^y = N^{-1} \sum_{i=1}^N \delta_{\xi_n^i}$  where  $\delta_\xi$  is the Dirac mass at  $\xi$ . Assume that the empirical approximation of  $\phi_{t+1}^y$  is  $\phi_{t+1}^N = N^{-1} \sum_{i=1}^N \delta_{\xi_{t+1}^i}$ , where  $\xi_{t+1}^{1:N}$  are  $N$  random variables. Substituting  $\phi_{t+1}^N$  into the recursion (2.5) and introducing the instrumental kernel (2.3), we obtain the mixture

$$\hat{\phi}_t^N(x_t) = \sum_{i=1}^N \tilde{\omega}_t(\xi_{t+1}^i) p_t^y(x_t | \xi_{t+1}^i) / \sum_{j=1}^N \tilde{\omega}_t(\xi_{t+1}^j). \quad (2.6)$$

Then, a particle approximation of (2.6) is obtained by sampling  $N$  conditionally i.i.d. ancestor indices  $I_{t+1}^{1:N} \stackrel{\text{i.i.d.}}{\sim} \text{Cat}(\{\tilde{\omega}_t(\xi_{t+1}^i) / \sum_{j=1}^N \tilde{\omega}_t(\xi_{t+1}^j)\}_{i=1}^N)$ , and then propagating each ancestor particle  $\xi_{t+1}^{I_{t+1}^i}$  according to the instrumental kernel (2.3). The final particle approximation is given by  $\phi_0^N = N^{-1} \sum_{i=1}^N \delta_{\xi_0^i}$ , where  $\xi_0^i \sim p_0(\cdot | \xi_1^{I_0^i})$ ,  $I_0^i \sim \text{Cat}(\{\tilde{\omega}_0(\xi_1^k) / \sum_{j=1}^N \tilde{\omega}_0(\xi_1^j)\}_{k=1}^N)$ . The sequence of distributions  $\{p_t\}_{t=0}^n$  approximating the marginals of the forward process initialized at  $p_0$  defines a path that bridges between  $p_n$  and the prior  $p_0$  such that the discrepancy between  $p_t$  and  $p_{t+1}$  is small. SMC samplers based on this path are robust to multi-modality and offer an interesting alternative to the geometric and tempering paths traditionally used in the SMC literature, see [Dai et al., 2022]. Our proposals  $\phi_t^y(x_t) \propto p_t(x_t) \bar{q}_{t|0}(\bar{x}_t | y)$  inherit the behavior of  $\{p_t\}_{t \in \mathbb{N}}$  and bridge the initial distribution  $\phi_0^y$  and posterior  $\phi_0^y$ . Indeed, as  $y$  is a noiseless observation of  $X_0 \sim p_0$ , we may consider  $\bar{\alpha}_t^{1/2} y + (1 - \bar{\alpha}_t)^{1/2} \varepsilon_t$ , with  $\varepsilon_t \sim \mathcal{N}(\mathbf{0}_{d_y}, \mathbf{I}_{d_y})$ , as a noisy observation of  $X_t \sim p_t$  and thus,  $\phi_t^y$  is the associated posterior. We illustrate this intuition by considering the following Gaussian mixture (GM) example. We assume that  $p_0(x_0) = \sum_{i=1}^M w_i \cdot \mathcal{N}(x_0; \mu_i, \mathbf{I}_{d_x})$  where  $M > 1$  and  $\{w_i\}_{i=1}^M$  are drawn uniformly on the simplex. The marginals of the forward process are available in closed form and are given by  $p_t(x_t) = \sum_{i=1}^M w_i \cdot \mathcal{N}(x_t; \bar{\alpha}_t^{1/2} \mu_i, \mathbf{I}_{d_x})$ , which shows that the discrepancy between  $p_t$  and  $p_{t+1}$  is small as long as  $\bar{\alpha}_t^{1/2} - \bar{\alpha}_{t+1}^{1/2} \approx 0$ . The posteriors  $\{\phi_t^y\}_{t \in [0:n]}$  are also available in closed form and displayed in Figure 1, which illustrates that our choice of potentials ensures that the discrepancy between consecutive posteriors is small. The idea of using the forward diffused observation to guide the observed part of the state, as we do here through  $\bar{q}_t(\bar{x}_t | y)$ , has been exploited in prior works but in a different way. For instance, in [Song et al., 2021c, Song et al., 2022] the observed part of the state is directly replaced by the forward noisy observation and, as it has been noted [Trippe et al., 2023], this introduces an irreducible bias. Instead, MCGdiff

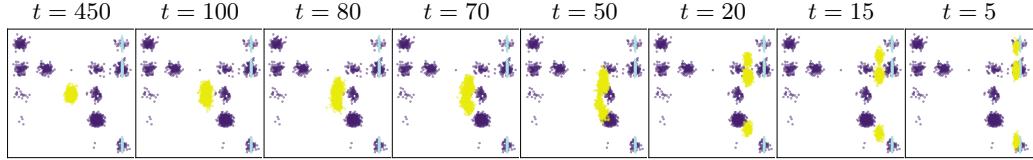


Figure 1: Display of samples from  $\phi_t^y(x_t) \propto p_t(x_t) \bar{q}_{t|0}(\bar{x}_t|y)$  for the GM prior. In yellow the samples from  $\phi_t^y$ , in purple those from the prior and in light blue those from the posterior  $\phi_0^y$ .  $n$  is set to 500.

weights the backward process by the density of the forward one conditioned on  $y$ , resulting in a natural and consistent algorithm.

We now establish the convergence of MCGdiff with a general sequence of *potentials*  $\{g_s^y\}_{s=1}^n$ . We consider the following assumption on the sequence of potentials  $\{g_t^y\}_{t=1}^n$ .

$$(\mathbf{A1}) \quad \sup_{x \in \mathbb{R}^{d_x}} \bar{p}_0(y|x)/g_1^y(x) < \infty \text{ and } \sup_{x \in \mathbb{R}^{d_x}} \int g_t^y(x_t) p_t(x_t|x) dx_t / g_{t+1}^y(x) < \infty \text{ for all } t \in [1 : n-1].$$

The following exponential deviation inequality is standard and is a direct application of [Douc et al., 2014, Theorem 10.17]. In particular, it implies a  $\mathcal{O}(1/\sqrt{N})$  bound on the mean squared error  $\|\phi_0^N(h) - \phi_0^y(h)\|_2$ .

**Proposition 2.1.** *Assume (A1). There exist constants  $c_{1,n}, c_{2,n} \in (0, \infty)$  such that, for all  $N \in \mathbb{N}$ ,  $\varepsilon > 0$  and bounded function  $h : \mathbb{R}^{d_x} \mapsto \mathbb{R}$ ,  $\mathbb{P}[|\phi_0^N(h) - \phi_0^y(h)| \geq \varepsilon] \leq c_{1,n} \exp(-c_{2,n} N \varepsilon^2 / \|h\|_\infty^2)$  where  $|h|_\infty := \sup_{x \in \mathbb{R}^{d_x}} |h(x)|$ .*

We also furnish our estimator with an explicit non-asymptotic bound on its bias. Define  $\Phi_0^N = \mathbb{E}[\phi_0^N]$  where  $\phi_0^N = N^{-1} \sum_{i=1}^N \delta_{\xi_i^N}$  is the particle approximation produced by Algorithm 1 and the expectation is with respect to the law of  $(\xi_{1:n}^{1:N}, I_{0:n-1}^{1:N})$ . Define for all  $t \in [1 : n]$ ,  $\phi_t^*(x_t) \propto p_t(x_t) \int \delta_y(d\bar{x}_0) p_{0|t}(x_0|x_t) d\bar{x}_0$ , where  $p_{0|t}(x_0|x_t) := \int \left\{ \prod_{s=0}^{t-1} p_s(x_s|x_{s+1}) \right\} dx_{1:t-1}$ .

**Proposition 2.2.** *It holds that*

$$\text{KL}(\phi_0^y \parallel \Phi_0^N) \leq C_{0:n}^y (N-1)^{-1} + D_{0:n}^y N^{-2}, \quad (2.7)$$

where  $D_{0:n}^y > 0$ ,  $C_{0:n}^y := \sum_{t=1}^n \int \frac{\mathcal{Z}_t/\mathcal{Z}_0}{g_t^y(z_t)} \left\{ \int \delta_y(d\bar{x}_0) p_{0|t}(x_0|z_t) d\bar{x}_0 \right\} \phi_t^*(dz_t)$  and  $\mathcal{Z}_t := \int g_t^y(x_t) p_t(dx_t)$  for all  $t \in [1 : n]$  and  $\mathcal{Z}_0 := \int \delta_y(d\bar{x}_0) p_0(x_0) d\bar{x}_0$ . If furthermore (A1) holds then both  $C_{0:n}^y$  and  $D_{0:n}^y$  are finite.

The proof of Proposition 2.2 is postponed to Appendix B.1. (A1) is an assumption on the equivalent of the weights  $\{\tilde{\omega}_t\}_{t=0}^n$  with a general sequence of potentials  $\{g_t^y\}_{t=1}^n$  and is not restrictive as it can be satisfied by setting for example  $g_s^y(x_s) = \bar{q}_{s|0}(\bar{x}_s|y) + \delta$  where  $\delta > 0$ . The resulting algorithm is then only a slight modification of the one described above, see Appendix B.1 for more details. It is also worth noting that Proposition 2.2 combined with Pinsker's inequality implies that the bias of MCGdiff goes to 0 with the number of particle samples  $N$  for fixed  $n$ . We have chosen to present a bound in Kullback–Leibler (KL) divergence, inspired by [Andrieu et al., 2018, Huggins and Roy, 2019], as it allows an explicit dependence on the modeling choice  $\{g_s^y\}_{s=1}^n$ , see Lemma B.2. Finally, unlike the theoretical guarantees established for SMCdiff in [Trippe et al., 2023], proving the asymptotic exactness of our methodology w.r.t. to the generative model posterior does not require having  $p_{s+1}(x_{s+1})p_s(x_s|x_{s+1}) = p_s(x_s)q_{s+1}(x_{s+1}|x_s)$  for all  $s \in [0 : n-1]$ , which does not hold in practice.

**Noisy case.** We consider the case  $\sigma_y > 0$ . The posterior density is given by  $\phi_0^y(x_0) \propto g_0^y(\bar{x}_0) p_0(x_0)$ , where  $g_0^y(x_0) := \mathcal{N}(y; \bar{x}_0, \sigma_y^2 \mathbf{I}_{d_y})$ . In what follows, assume that there exists  $\tau \in [1 : n]$  such that  $\sigma^2 = (1 - \bar{\alpha}_\tau)/\bar{\alpha}_\tau$ . We denote  $\tilde{y}_\tau = \bar{\alpha}_\tau^{1/2} y$ . We can then write that

$$g_0^y(\bar{x}_0) = \bar{\alpha}_\tau^{1/2} \cdot \mathcal{N}(\tilde{y}_\tau; \bar{\alpha}_\tau^{1/2} x_0, (1 - \bar{\alpha}_\tau) \cdot \mathbf{I}_{d_y}) = \bar{\alpha}_\tau^{1/2} \cdot \bar{q}_{\tau|0}(\tilde{y}_\tau|\bar{x}_0), \quad (2.8)$$

which hints that the likelihood function  $g_0^y$  is closely related to the forward process (1.1). We may then write the posterior  $\phi_0^y(x_0)$  as  $\phi_0^y(x_0) \propto \bar{q}_{\tau|0}(\tilde{y}_\tau|\bar{x}_0) p_0(x_0) \propto \int \delta_{\tilde{y}_\tau}(d\bar{x}_\tau) q_{\tau|0}(x_\tau|x_0) p_0(x_0) d\bar{x}_\tau$ . Next, assume that the

forward process (1.1) is the reverse of the backward one (1.3), i.e. that

$$\mathbf{p}_t(x_t)q_{t+1}(x_{t+1}|x_t) = \mathbf{p}_{t+1}(x_{t+1})p_t(x_t|x_{t+1}), \quad \forall t \in [0 : n - 1]. \quad (2.9)$$

This is similar to the assumption made in **SMCdiff** [Trippe et al., 2023]. Then, it is easily seen that it implies  $\mathbf{p}_0(x_0)q_{\tau|0}(x_\tau|x_0) = \mathbf{p}_\tau(x_\tau)p_{0|\tau}(x_0|x_\tau)$  and thus

$$\phi_0^y(x_0) = \int p_{0|\tau}(x_0|x_\tau) \delta_{\tilde{y}_\tau}(\mathrm{d}\bar{x}_\tau) \mathbf{p}_\tau(x_\tau) \mathrm{d}\underline{x}_\tau / \int \delta_{\tilde{y}_\tau}(\mathrm{d}\bar{z}_\tau) \mathbf{p}_\tau(z_\tau) \mathrm{d}\underline{z}_\tau = \int p_{0|\tau}(x_0|\tilde{y}_\tau \cap \underline{x}_\tau) \phi_\tau^{\tilde{y}_\tau}(\mathrm{d}\underline{x}_\tau), \quad (2.10)$$

where  $\phi_\tau^{\tilde{y}_\tau}(\underline{x}_\tau) \propto \mathbf{p}_\tau(\tilde{y}_\tau \cap \underline{x}_\tau)$ . (2.10) highlights that solving the inverse problem (2.1) with  $\sigma_y > 0$  is equivalent to solving an inverse problem on the intermediate state  $X_\tau \sim \mathbf{p}_\tau$  with *noiseless* observation  $\tilde{y}_\tau$  of the  $d_y$  top coordinates and then propagating the resulting posterior back to time 0 with the backward kernel  $p_{0|\tau}$ . The assumption (2.9) does not always hold in realistic settings. Therefore, while (2.10) also holds only approximately in practice, we can still use it as inspiration for designing potentials when the assumption is not valid. Consider then  $\{g_t^y\}_{t=\tau}^n$  and sequence of probability measures  $\{\phi_t^y\}_{t=\tau}^n$  defined for all  $t \in [\tau : n]$  as  $\phi_t^y(x_t) \propto g_t^y(x_t) \mathbf{p}_t(x_t)$ , where  $g_t^y(x_t) := \mathcal{N}(x_t; \tilde{y}_\tau, (1 - (1 - \kappa)\bar{\alpha}_t/\bar{\alpha}_\tau)I_{d_y})$ ,  $\kappa \geq 0$ . In the case of  $\kappa = 0$ , we have  $g_t^y(x_t) = \bar{q}_{t|\tau}(\bar{x}_t|\tilde{y}_\tau)$  for  $t \in [\tau + 1 : n]$  and  $\phi_\tau^y = \phi_{\tau|0}^y$ . The recursion (2.5) holds for  $t \in [\tau : n]$  and assuming  $\kappa > 0$ , we find that  $\phi_0^y(x_0) \propto g_0^y(x_0) \int g_\tau^y(x_\tau)^{-1} p_{0|\tau}(x_0|x_\tau) \phi_\tau^y(\mathrm{d}x_\tau)$ , which resembles the recursion (2.10). In practice we take  $\kappa$  to be small in order to mimick the Dirac delta mass at  $\bar{x}_\tau$  in (2.10). Having a particle approximation  $\phi_\tau^N = N^{-1} \sum_{i=1}^N \delta_{\xi_\tau^i}$  of  $\phi_\tau^y$  by adapting Algorithm 1, we estimate  $\phi_0^y$  with  $\phi_0^N = \sum_{i=1}^N \omega_0^i \delta_{\xi_0^i}$  where  $\xi_0^i \sim p_{0|\tau}(\cdot|\xi_\tau^i)$  and  $\omega_0^i \propto g_0^y(\xi_0^i)$ . In the next section we extend this methodology to general linear Gaussian observation models. Finally, note that (2.10) allows us to extend **SMCdiff** to handle noisy inverse problems in a principled manner. This is detailed in Appendix A.

## 2.1 Extension to general linear inverse problems

Consider  $Y = AX + \sigma_y \varepsilon$  where  $A \in \mathbb{R}^{d_y \times d_x}$ ,  $\varepsilon \sim \mathcal{N}(0_{d_y}, I_{d_y})$  and  $\sigma_y \geq 0$  and the singular value decomposition (SVD)  $A = US\bar{V}^T$ , where  $\bar{V} \in \mathbb{R}^{d_x \times d_y}$ ,  $U \in \mathbb{R}^{d_y \times d_y}$  are two orthonormal matrices, and  $S \in \mathbb{R}^{d_y \times d_y}$  is diagonal. For simplicity, it is assumed that the singular values satisfy  $s_1 > \dots > s_{d_y} > 0$ . Set  $b = d_x - d_y$ . Let  $V \in \mathbb{R}^{d_x \times b}$  be an orthonormal matrix of which the columns complete those of  $\bar{V}$  into an orthonormal basis of  $\mathbb{R}^{d_x}$ , i.e.  $V^T V = I_b$  and  $V^T \bar{V} = 0_{b,d_y}$ . We define  $V = [\bar{V}, \underline{V}] \in \mathbb{R}^{d_x \times d_x}$ . In what follows, for a given  $\mathbf{x} \in \mathbb{R}^{d_x}$  we write  $\bar{\mathbf{x}} \in \mathbb{R}^{d_y}$  for its top  $d_y$  coordinates and  $\underline{\mathbf{x}} \in \mathbb{R}^b$  for the remaining coordinates. Setting  $\mathbf{X} := V^T X$  and  $\mathbf{Y} := S^{-1} U^T Y$  and multiplying the measurement equation by  $S^{-1} U^T$  yields

$$\mathbf{Y} = \bar{\mathbf{X}} + \sigma_y S^{-1} \tilde{\varepsilon}, \quad \tilde{\varepsilon} \sim \mathcal{N}(0, I_{d_y}).$$

In this section, we focus on solving this linear inverse problem in the orthonormal basis defined by  $V$  using the methodology developed in the previous sections. This prompts us to define the diffusion based generative model in this basis. As  $V$  is an orthonormal matrix, the law of  $\mathbf{X}_0 = V^T X_0$  is  $\mathbf{p}_0(\mathbf{x}_0) := \mathbf{p}_0(V\mathbf{x}_0)$ . By definition of  $\mathbf{p}_0$  and the fact that  $\|V\mathbf{x}\|_2 = \|\mathbf{x}\|_2$  for all  $\mathbf{x} \in \mathbb{R}^{d_x}$  we have that

$$\mathbf{p}_0(\mathbf{x}_0) = \int p_0(V\mathbf{x}_0|\mathbf{x}_1) \left\{ \prod_{s=1}^{n-1} p_s(\mathrm{d}\mathbf{x}_s|\mathbf{x}_{s+1}) \right\} \mathbf{p}_n(\mathrm{d}\mathbf{x}_n) = \int \lambda_0(\mathbf{x}_0|\mathbf{x}_1) \left\{ \prod_{s=1}^{n-1} \lambda_s(\mathrm{d}\mathbf{x}_s|\mathbf{x}_{s+1}) \right\} \mathbf{p}_n(\mathrm{d}\mathbf{x}_n)$$

where for all  $s \in [1 : n]$ ,  $\lambda_{s-1}(\mathbf{x}_{s-1}|\mathbf{x}_s) := \mathcal{N}(\mathbf{x}_{s-1}; \mathbf{m}_s(\mathbf{x}_s), \sigma_s^2 I_{d_x})$ , where  $\mathbf{m}_s(\mathbf{x}_s) := V^T \mathbf{m}_s(V\mathbf{x}_s)$ . The transition kernels  $\{\lambda_s\}_{s=0}^{n-1}$  thus define a diffusion based model in the basis  $V$ . In what follows we write  $\bar{\mathbf{m}}_s(\mathbf{x}_s)$  for the first  $d_y$  coordinates of  $\mathbf{m}_s(\mathbf{x}_s)$  and  $\underline{\mathbf{m}}_s(\mathbf{x}_s)$  the last  $b$  coordinates. We denote by  $\mathbf{p}_s$  the time  $s$  marginal of the backward process.

**Noiseless.** In this case the target posterior is  $\phi_0^y(\mathbf{x}_0) \propto \mathbf{p}_0(y \cap \underline{\mathbf{x}}_0)$ . The extension of algorithm 1 is straight forward; it is enough to replace  $y$  with  $\mathbf{y}$  ( $= S^{-1} U^T y$ ) and the backward kernels  $\{p_t\}_{t=0}^{n-1}$  with  $\{\lambda_t\}_{t=0}^{n-1}$ .

**Noisy.** The posterior density is then  $\phi_0^y(\mathbf{x}_0) \propto g_0^y(\bar{\mathbf{x}}_0) \mathbf{p}_0(\mathbf{x}_0)$ , where

$$g_0^y(\bar{\mathbf{x}}_0) = \prod_{i=1}^{d_y} \mathcal{N}(\mathbf{y}[i]; \bar{\mathbf{x}}_0[i], (\sigma_y/s_i)^2). \quad (2.11)$$

As in Section 2, assume that there exists  $\{\tau_i\}_{i=1}^{d_y} \subset [1 : n]$  such that  $\bar{\alpha}_{\tau_i} \sigma_y^2 = (1 - \bar{\alpha}_{\tau_i}) s_i^2$  and define for all  $i \in [1 : d_y]$ ,  $\tilde{\mathbf{y}}_i := \bar{\alpha}_{\tau_i}^{1/2} \mathbf{y}[i]$ . Then we can write the potential  $g_0^y$  in a similar fashion to (2.8) as the product

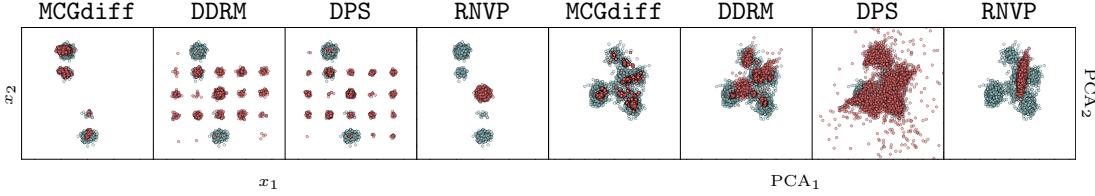


Figure 2: The first and last four columns correspond respectively to GM with  $(d_x, d_y) = (800, 1)$  and FM with  $(d_x, d_y) = (10, 1)$ . The blue and red dots represent respectively samples from the exact posterior and those generated by each of the algorithms used (names on top).

$d$	$d_y$	MCGdiff	DDRM	DPS	RNVP	$d$	$d_y$	MCGdiff	DDRM	DPS	RNVP
80	1	<b><math>1.39 \pm 0.45</math></b>	$5.64 \pm 1.10$	$4.98 \pm 1.14$	$6.86 \pm 0.88$	6	1	<b><math>1.95 \pm 0.43</math></b>	$4.20 \pm 0.78$	$5.43 \pm 1.05$	$6.16 \pm 0.65$
80	2	<b><math>0.67 \pm 0.24</math></b>	$7.07 \pm 1.35$	$5.10 \pm 1.23$	$7.79 \pm 1.50$	6	3	<b><math>0.73 \pm 0.33</math></b>	$2.20 \pm 0.67$	$3.47 \pm 0.78$	$4.70 \pm 0.90$
80	4	<b><math>0.28 \pm 0.14</math></b>	$7.81 \pm 1.48$	$4.28 \pm 1.26$	$7.95 \pm 1.61$	6	5	<b><math>0.41 \pm 0.12</math></b>	$0.91 \pm 0.43$	$2.07 \pm 0.63$	$3.52 \pm 0.93$
800	1	<b><math>2.40 \pm 1.00</math></b>	$7.44 \pm 1.15$	$6.49 \pm 1.16$	$7.74 \pm 1.34$	10	1	<b><math>2.45 \pm 0.42</math></b>	$3.82 \pm 0.64$	$4.30 \pm 0.91$	$6.04 \pm 0.38$
800	2	<b><math>1.31 \pm 0.60</math></b>	$8.95 \pm 1.12$	$6.88 \pm 1.01$	$8.75 \pm 1.02$	10	3	<b><math>1.07 \pm 0.26</math></b>	$4.94 \pm 0.87$	$5.38 \pm 0.84$	$5.91 \pm 0.64$
800	4	<b><math>0.47 \pm 0.19</math></b>	$8.39 \pm 1.48$	$5.51 \pm 1.18$	$7.81 \pm 1.63$	10	5	<b><math>0.71 \pm 0.12</math></b>	$2.32 \pm 0.74$	$3.74 \pm 0.77$	$5.11 \pm 0.69$

Table 1: Sliced Wasserstein for the GM (left) and FM (right) case.

of forward processes from time 0 to each time step  $\tau_i$ , i.e.  $g_0^{\mathbf{y}}(\mathbf{x}_0) = \prod_{i=1}^{d_y} \bar{\alpha}_{\tau_i}^{1/2} \mathcal{N}(\tilde{\mathbf{y}}_i; \bar{\alpha}_{\tau_i}^{1/2} \mathbf{x}_0[i], (1 - \bar{\alpha}_{\tau_i}))$ . Writing the potential this way allows us to generalize (2.10) as follows. Denote for  $\ell \in [1 : d_x]$ ,  $\mathbf{x}^{\setminus \ell} \in \mathbb{R}^{d_x - 1}$  the vector  $\mathbf{x}$  with its  $\ell$ -th coordinate removed. Define

$$\phi_{\tau_1:n}^{\tilde{\mathbf{y}}}(\mathrm{d}\mathbf{x}_{\tau_1:n}) \propto \left\{ \prod_{i=1}^{d_y-1} \lambda_{\tau_i|\tau_{i+1}}(\mathbf{x}_{\tau_i} | \mathbf{x}_{\tau_{i+1}}) \delta_{\tilde{\mathbf{y}}_i}(\mathrm{d}\mathbf{x}_{\tau_i}[i]) \mathrm{d}\mathbf{x}_{\tau_i}^{\setminus i} \right\} \mathfrak{p}_{\tau_{d_y}}(\mathbf{x}_{\tau_{d_y}}) \delta_{\tilde{\mathbf{y}}_{d_y}}(\mathrm{d}\mathbf{x}_{\tau_{d_y}}[d_y]) \mathrm{d}\mathbf{x}_{\tau_{d_y}}^{\setminus d_y},$$

which corresponds to the posterior of a noiseless inverse problem on the joint states  $\mathbf{X}_{\tau_1:n} \sim \mathfrak{p}_{\tau_1:n}$  with noiseless observations  $\tilde{\mathbf{y}}_{\tau_i}$  of  $\mathbf{X}_{\tau_i}$  for all  $i \in [1 : d_y]$ .

**Proposition 2.3.** Assume that  $\mathfrak{p}_{s+1}(\mathbf{x}_{s+1})\lambda_s(\mathbf{x}_s | \mathbf{x}_{s+1}) = \mathfrak{p}_s(\mathbf{x}_s)q_{s+1}(\mathbf{x}_{s+1} | \mathbf{x}_s)$  for all  $s \in [0 : n - 1]$ . Then it holds that  $\phi_0^{\mathbf{y}}(\mathbf{x}_0) \propto \int \lambda_{0|\tau_1}(\mathbf{x}_0 | \mathbf{x}_{\tau_1}) \phi_{\tau_1:n}^{\tilde{\mathbf{y}}}(\mathrm{d}\mathbf{x}_{\tau_1:n})$ .

The proof of Proposition 2.3 is given in Appendix B.2. We have shown that sampling from  $\phi_0^{\mathbf{y}}$  is equivalent to sampling from  $\phi_{\tau_1:n}^{\tilde{\mathbf{y}}}$  then propagating the final state  $\mathbf{X}_{\tau_1}$  to time 0 according to  $\lambda_{0|\tau_1}$ . Therefore, as in (2.8), we define  $\{g_t^y\}_{t=\tau}^n$  and  $\{\phi_t^y\}_{t=\tau}^n$  for all  $t \in [\tau_1 : n]$  by  $\phi_t^y(\mathbf{x}_t) \propto g_t^y(\mathbf{x}_t) p_t(\mathbf{x}_t)$  and  $g_t^y(\mathbf{x}_t) := \prod_{i=1}^{d_y} \mathcal{N}(\mathbf{x}_t; \tilde{\mathbf{y}}_i, 1 - (1 - \kappa)\bar{\alpha}_t/\bar{\alpha}_{\tau_1})$ ,  $\kappa > 0$ . We obtain a particle approximation of  $\phi_{\tau_1}^y$  using a particle filter with proposal kernel and weight function  $\lambda_t^y(\mathbf{x}_t | \mathbf{x}_{t+1}) \propto g_t^y(\mathbf{x}_t) p_t(\mathbf{x}_t | \mathbf{x}_{t+1})$ ,  $\tilde{\omega}_t(\mathbf{x}_{t+1}) = \int g_t^y(\mathbf{x}_t) p_t(\mathrm{d}\mathbf{x}_t | \mathbf{x}_{t+1}) / g_{t+1}^y(\mathbf{x}_{t+1})$ , which are both available in closed form.

### 3 Numerics

The focus of this work is on providing an algorithm that consistently approximates the posterior distribution of a linear inverse problem with Gaussian noise. A prerequisite for quantitative evaluation in ill-posed inverse problems in a Bayesian setting is to have access to samples of the posterior distribution. This generally requires having at least an unnormalized proxy of the posterior density, so that one can run MCMC samplers such as the No U-turn sampler (NUTS) [Hoffman and Gelman, 2011]. Therefore, this section focus on mixture models of two types of basis distribution, the Gaussian and the Funnel distributions. We then present a brief illustration of MCGdiff on image data. However, in this setting, the actual posterior distribution is unknown and the main goal is to explore the potentially multimodal posterior distribution, which makes a comparison with a "real image" meaningless. Therefore, metrics such as Fréchet Inception Distance (FID) and LPIPS score, which require comparison to a ground truth, are not useful for evaluating Bayesian reconstruction methods in such settings.<sup>1</sup>

**Mixture Models:** We refer to the Funnel mixture prior as FM prior (see appendix B.3 for the definition). For GM prior, we consider a mixture of 25 components with pre-established means and variances. For FM prior, we consider a mixture of 20 components consisting of rotated and translated funnel distributions. For a

<sup>1</sup>The code for the experiments is available at <https://anonymous.4open.science/r/mcgdiff/README.md>.

given pair  $(d_x, d_y)$ , we sample a prior distribution by randomly sampling the weights of the mixture and for the FM case the translation and rotation of each component. We then randomly sample measurement models  $(y, A, \sigma_y) \in \mathbb{R}^{d_y} \times \mathbb{R}^{d_y \times d_x} \times [0, 1]$ . For each pair of prior distribution and measurement model, we generate  $10^4$  samples from MCGdiff, DPS, DDRM, RNVP, and from the posterior either analytically (GM) or using NUTS (FM). We calculate for each algorithm the sliced Wasserstein (SW) distance between the resulting samples and the posterior samples. Table 1 shows the CLT 95% confidence intervals obtained over 20 seeds. Figure 2 illustrate the samples for the different algorithms for a given seed. We see that MCGdiff outperforms all the other algorithms in each setting tested. The complete details of the numerical experiments performed in this section is available in appendix B.3 as well as an additional visualisations.

### 3.0.1 Image datasets

We now present samples from MCGdiff in different image dataset and different kinds of inverse problems.

**Super Resolution** We start by super resolution. We set  $\sigma_y = 0.05$  for all the datasets and  $\zeta_{\text{coeff}} = 0.1$  for DPS . We use 100 steps of DDIM with  $\eta = 1$ . The results are shown in Figure 3. We use a downsampling ratio of 4 for the CIFAR-10 dataset, 8 for both Flowers and Cats datasets and 16 for the others. The dimension of the datasets are recalled in table 2. We display in fig. 3 samples from MCGdiff, DPSand DDRMover several different image datasets (table 2). For each algorithm, we generate 1000 samples and we show the pair of samples that are the furthest apart in  $L^2$  norm from each other in the pool of samples. For MCGdiff we ran several parallel particle filters with  $N = 64$  to generate 1000 samples.

	CIFAR-10	Flowers	Cats	Bedroom	Church	CelebaHQ
$(W, H, C)$	$(32, 32, 3)$	$(64, 64, 3)$	$(128, 128, 3)$	$(256, 256, 3)$	$(256, 256, 3)$	$(256, 256, 3)$

Table 2: The datasets used for the inverse problems over image datasets.

**Gaussian 2D deblurring** We consider a Gaussian 2D square kernel with sizes  $(w/6, h/6)$  and standard deviation  $w/30$  where  $(w, h)$  are the width and height of the image. We set  $\sigma_y = 0.1$  for all the datasets and  $\zeta_{\text{coeff}} = 0.1$  for DPS . We use 100 steps of DDIM with  $\eta = 1$ . We display in fig. 4 samples from MCGdiff, DPSand DDRMover several different image datasets (table 2). For each algorithm, we generate 1000 samples and we show the pair of samples that are the furthest appart in  $L^2$  norm from each other in the pool of samples. For MCGdiff we ran several parallel particle filters with  $N = 64$  to generate 1000 samples.

**Inpainting on CelebA** We consider the inpainting problem on the CelebA dataset with several different masks in fig. 5. We show in fig. 6 the evolution of the particle cloud with  $s$ .

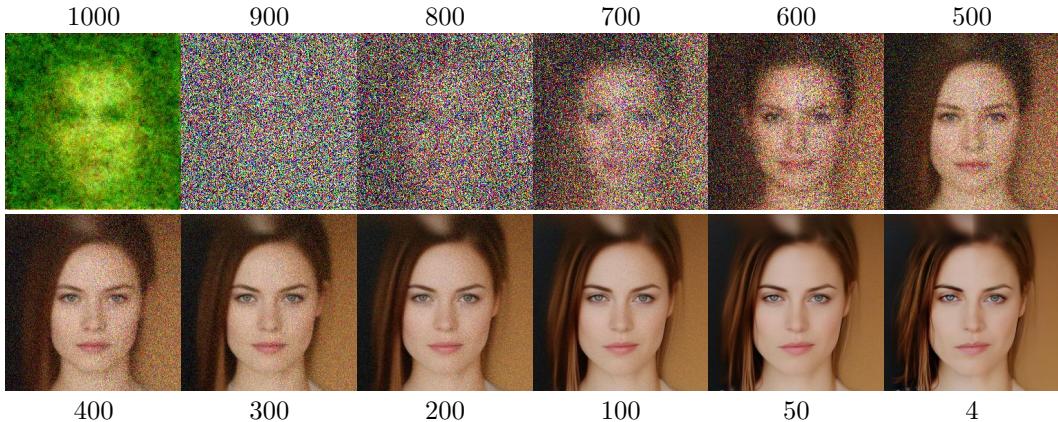


Figure 6: Evolution of the particle cloud for one of the masks. The numbers on top and bottom indicate the step  $s$  of the approximation.



Figure 3: Ratio 4 for CIFAR, 8 for flowers and Cats and 16 for CELEB

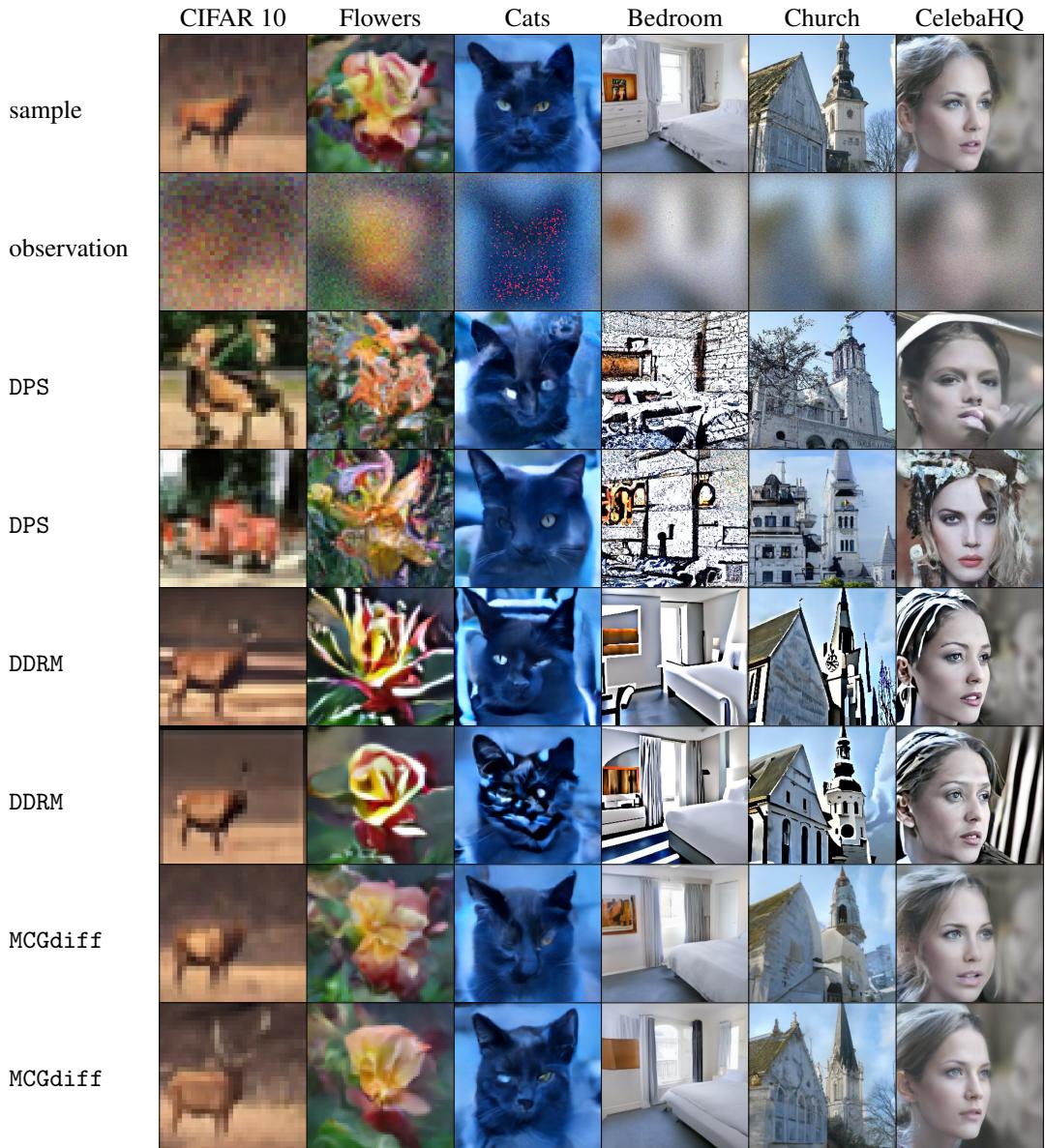


Figure 4



Figure 5: Inpainting with different masks on the CelebA test set.

## 4 Conclusion

In this paper, we present MCGdiff a novel method for solving Bayesian linear Gaussian inverse problems with SGM priors. We show that MCGdiff is theoretically grounded and provided numerical experiments that reflect the adequacy of MCGdiff in a Bayesian framework, as opposed to recent works. This difference is of the uttermost importance when the relevance of the generated samples is hard to verify, as in safety critical applications. MCGdiff is a first step towards robust approaches for addressing the challenges of Bayesian linear inverse problems with SGM priors.

**Acknowledgments** GC, YJEL and EM would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the program *The mathematical and statistical foundation of future data-driven engineering* when work on this paper was undertaken

## References

- [Andrieu et al., 2018] Andrieu, C., Lee, A., and Vihola, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842 – 872.
- [Arjomand Bigdeli et al., 2017] Arjomand Bigdeli, S., Zwicker, M., Favaro, P., and Jin, M. (2017). Deep mean-shift priors for image restoration. *Advances in Neural Information Processing Systems*, 30.
- [Benton et al., 2022] Benton, J., Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2022). From denoising diffusions to denoising markov models. *arXiv preprint arXiv:2211.03595*.
- [Besag et al., 1991] Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the institute of statistical mathematics*, 43:1–20.
- [Cappé et al., 2005] Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- [Cheney and Borden, 2009] Cheney, M. and Borden, B. (2009). *Fundamentals of radar imaging*. SIAM.
- [Chopin et al., 2020] Chopin, N., Papaspiliopoulos, O., et al. (2020). *An introduction to sequential Monte Carlo*, volume 4. Springer.
- [Chung et al., 2023] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.
- [Dai et al., 2022] Dai, C., Heng, J., Jacob, P. E., and Whiteley, N. (2022). An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, 117(539):1587–1600.

- [Dashti and Stuart, 2017] Dashti, M. and Stuart, A. M. (2017). The bayesian approach to inverse problems. In *Handbook of uncertainty quantification*, pages 311–428. Springer.
- [Dinh et al., 2017] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- [Douc et al., 2014] Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples*. CRC press.
- [Doucet et al., 2001] Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.
- [Elbakri and Fessler, 2002] Elbakri, I. A. and Fessler, J. A. (2002). Statistical image reconstruction for polyenergetic x-ray computed tomography. *IEEE transactions on medical imaging*, 21(2):89–99.
- [Fergus et al., 2006] Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. (2006). Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pages 787–794.
- [Figueiredo et al., 2007] Figueiredo, M. A., Bioucas-Dias, J. M., and Nowak, R. D. (2007). Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991.
- [González et al., 2009] González, R. C., Woods, R. E., and Masters, B. R. (2009). Digital image processing, third edition. *Journal of Biomedical Optics*, 14:029901.
- [Gui et al., 2021] Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*.
- [Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.
- [Hoffman and Gelman, 2011] Hoffman, M. and Gelman, A. (2011). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15.
- [Huggins and Roy, 2019] Huggins, J. H. and Roy, D. M. (2019). Sequential Monte Carlo as approximate sampling: bounds, adaptive resampling via  $\infty$ -ESS, and an application to particle Gibbs. *Bernoulli*, 25(1):584 – 622.
- [Idier, 2013] Idier, J. (2013). *Bayesian approach to inverse problems*. John Wiley & Sons.
- [Ivanov et al., 2018] Ivanov, O., Figurnov, M., and Vetrov, D. (2018). Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*.
- [Kaipio et al., 2000] Kaipio, J. P., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. (2000). Statistical inversion and monte carlo sampling methods in electrical impedance tomography. *Inverse problems*, 16(5):1487.
- [Kaltenbach et al., 2023] Kaltenbach, S., Perdikaris, P., and Koutsourelakis, P.-S. (2023). Semi-supervised invertible neural operators for bayesian inverse problems. *Computational Mechanics*, pages 1–20.
- [Kawar et al., ] Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*.
- [Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [Kingma et al., 2019] Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.
- [Kobyzev et al., 2020] Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979.

- [Lugmayr et al., 2022] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471.
- [Marnissi et al., 2017] Marnissi, Y., Zheng, Y., Chouzenoux, E., and Pesquet, J.-C. (2017). A variational bayesian approach for image restoration—application to image deblurring with poisson–gaussian noise. *IEEE Transactions on Computational Imaging*, 3(4):722–737.
- [Peng et al., 2021] Peng, J., Liu, D., Xu, S., and Li, H. (2021). Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784.
- [Pitt and Shephard, 1999] Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.
- [Sahlström and Tarvainen, 2023] Sahlström, T. and Tarvainen, T. (2023). Utilizing variational autoencoders in the bayesian inverse problem of photoacoustic tomography. *SIAM Journal on Imaging Sciences*, 16(1):89–110.
- [Shin and Choi, 2023] Shin, H. and Choi, M. (2023). Physics-informed variational inference for uncertainty quantification of stochastic differential equations. *Journal of Computational Physics*, page 112183.
- [Sohl-Dickstein et al., 2015] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.
- [Song et al., 2021a] Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.
- [Song et al., 2021b] Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021b). Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428.
- [Song et al., 2022] Song, Y., Shen, L., Xing, L., and Ermon, S. (2022). Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*.
- [Song et al., 2021c] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021c). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
- [Stuart, 2010] Stuart, A. M. (2010). Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559.
- [Su et al., 2022] Su, J., Xu, B., and Yin, H. (2022). A survey of deep learning approaches to image restoration. *Neurocomputing*, 487:46–65.
- [Trippe et al., 2023] Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*.
- [Vlaardingerbroek and Boer, 2013] Vlaardingerbroek, M. T. and Boer, J. A. (2013). *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media.
- [Wan et al., 2021] Wan, Z., Zhang, J., Chen, D., and Liao, J. (2021). High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701.
- [Wei et al., 2022] Wei, X., van Gorp, H., Gonzalez-Carabarin, L., Freedman, D., Eldar, Y. C., and van Sloun, R. J. (2022). Deep unfolding with normalizing flow priors for inverse problems. *IEEE Transactions on Signal Processing*, 70:2962–2971.
- [Wu et al., 2023] Wu, L., Trippe, B. L., Naesseth, C. A., Blei, D. M., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models. *arXiv preprint arXiv:2306.17775*.
- [Xiang et al., 2023] Xiang, H., Zou, Q., Nawaz, M. A., Huang, X., Zhang, F., and Yu, H. (2023). Deep learning for image inpainting: A survey. *Pattern Recognition*, 134:109046.

- [Yeh et al., 2018] Yeh, R. A., Lim, T. Y., Chen, C., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. (2018). Image restoration with deep generative models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6772–6776. IEEE.
- [Yu et al., 2018] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514.
- [Zeng et al., 2022] Zeng, Y., Fu, J., Chao, H., and Guo, B. (2022). Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*.
- [Zhang et al., 2023] Zhang, G., Ji, J., Zhang, Y., Yu, M., Jaakkola, T., and Chang, S. (2023). Towards coherent image inpainting using denoising diffusion implicit models. *arXiv preprint arXiv:2304.03322*.
- [Zheng et al., 2019] Zheng, C., Cham, T.-J., and Cai, J. (2019). Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447.
- [Zhihang et al., 2023] Zhihang, X., Yingzhi, X., and Qifeng, L. (2023). A domain-decomposed vae method for bayesian inverse problems. *arXiv preprint arXiv:2301.05708*.

## A SMCdiff extension

The identity (2.10) allows us to extend SMCdiff [Trippe et al., 2023] to handle noisy inverse problems as we now show. We have that

$$\begin{aligned}\phi_{\tau}^{\tilde{y}_{\tau}}(\underline{x}_{\tau}) &= \frac{\int p_{\tau}(\tilde{y}_{\tau} \cap \underline{x}_{\tau} | x_{\tau+1}) \left\{ \prod_{s=\tau+1}^{n-1} p_s(dx_s | x_{s+1}) \right\} \mathbf{p}_n(dx_n)}{\int \mathbf{p}_{\tau}(\tilde{y}_{\tau} \cap \underline{z}_{\tau}) d\underline{z}_{\tau}} \\ &= \int b_{\tau:n}^{\tilde{y}_{\tau}}(\underline{x}_{\tau:n} | \bar{x}_{\tau+1:n}) f_{\tau+1:n}^{\tilde{y}_{\tau}}(d\bar{x}_{\tau+1:n}) d\underline{x}_{\tau+1:n},\end{aligned}$$

where

$$\begin{aligned}b_{\tau:n}(\underline{x}_{\tau:n} | \bar{x}_{\tau+1:n}) &= \frac{p_{\tau}(\tilde{y}_{\tau} \cap \underline{x}_{\tau} | x_{\tau+1}) \left\{ \prod_{s=\tau+1}^{n-1} \underline{p}_s(x_s | x_{s+1}) \bar{p}_s(\bar{x}_s | x_{s+1}) \right\} \underline{\mathbf{p}}_n(\underline{x}_n)}{\mathsf{L}_{\tau:n}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n})}, \\ f_{\tau+1:n}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n}) &= \frac{\mathsf{L}_{\tau:n}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n})}{\int \mathbf{p}_{\tau}(\tilde{y}_{\tau} \cap \underline{z}_{\tau}) d\underline{z}_{\tau}},\end{aligned}$$

and

$$\mathsf{L}_{\tau:n}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n}) = \int p_{\tau}(\tilde{y}_{\tau} \cap \underline{z}_{\tau} | \bar{x}_{\tau+1} \cap \underline{z}_{\tau+1}) \left\{ \prod_{s=\tau+1}^{n-1} \underline{p}_s(d\underline{z}_s | \bar{x}_{s+1} \cap \underline{z}_{s+1}) \bar{p}_s(\bar{x}_s | \bar{x}_{s+1} \cap \underline{z}_{s+1}) \right\} \underline{\mathbf{p}}_n(d\underline{z}_n).$$

Next, (2.9) implies that

$$\begin{aligned}\int \mathbf{p}_{s+1}(\bar{x}_{s+1} \cap \underline{z}_{s+1}) \underline{p}_s(d\underline{z}_s | \bar{x}_{s+1} \cap \underline{z}_{s+1}) \bar{p}_s(\bar{x}_s | \bar{x}_{s+1} \cap \underline{z}_{s+1}) d\underline{z}_{s:s+1} = \\ \int \mathbf{p}_s(\bar{x}_s \cap \underline{z}_s) \bar{q}_{s+1}(\bar{x}_{s+1} | \bar{x}_s) q_{s+1}(z_{s+1} | z_s) d\underline{z}_{s:s+1},\end{aligned}$$

and applied repeatedly, we find that

$$\mathsf{L}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n}) = \int \mathbf{p}_{\tau}(\tilde{y}_{\tau} \cap \underline{x}_{\tau}) d\underline{x}_{\tau} \cdot \int \delta_{\tilde{y}_{\tau}}(d\bar{x}_{\tau}) \prod_{s=\tau+1}^n \bar{q}_s(\bar{x}_s | \bar{x}_{s-1}).$$

and thus,  $f_{\tau:n}^{\tilde{y}_{\tau}}(\bar{x}_{\tau+1:n}) = \int \delta_{\tilde{y}_{\tau}}(d\bar{x}_{\tau}) \prod_{s=\tau+1}^n \bar{q}_s(\bar{x}_s | \bar{x}_{s-1})$ . In order to approximate  $\phi_{\tau}^{\tilde{y}_{\tau}}$  we first diffuse the noised observation up to time  $n$ , resulting in  $\bar{x}_{\tau+1:n}$ , and then estimate  $b_{\tau+1:n}^{\tilde{y}_{\tau}}(\cdot | \bar{x}_{\tau+1:n})$  using a particle filter with  $\underline{p}_s(x_s | x_{s+1})$  as transition kernel at step  $s \in [\tau+1 : n]$  and  $g_s : \underline{z}_s \mapsto \bar{p}_{s-1}(\bar{x}_{s-1} | \bar{x}_s \cap \underline{z}_s)$  as potential, similarly to SMCdiff.

## B Proofs

### B.1 Proof of Proposition 2.2

#### Preliminary definitions.

We preface the proof with notations and definitions of a few quantities that will be used throughout.

For a probability measure  $\mu$  and  $f$  a bounded measurable function, we write  $\mu(f) := \int f(x)\mu(dx)$  the expectation of  $f$  under  $\mu$  and if  $K(dx|z)$  is a transition kernel we write  $K(f)(z) := \int f(x)K(dx|z)$ .

Define the *smoothing* distribution

$$\phi_{0:n}^y(dx_{0:n}) \propto \delta_y(d\bar{x}_0) \mathbf{p}_{0:n}(x_{0:n}) d\underline{x}_0 dx_{1:n}, \quad (\text{B.1})$$

which admits the posterior  $\phi_0^y$  as time 0 marginal. Its particle estimate known as the *poor man smoother* is given by

$$\phi_{0:n}^N(dx_{0:n}) = N^{-1} \sum_{k_{0:n} \in [1:N]^{n+1}} \delta_{y \cap \xi_0^{k_0}}(dx_0) \prod_{s=1}^n \mathbb{1}\{k_s = I_s^{k_{s-1}}\} \delta_{\xi_s^{k_s}}(dx_s). \quad (\text{B.2})$$

We also let  $\Phi_{0:n}^N$  be the probability measure defined for any  $B \in \mathcal{B}(\mathbb{R}^{\mathbf{d}_x})^{\otimes n+1}$  by

$$\Phi_{0:n}^N(B) = \mathbb{E}[\phi_{0:n}^N(B)],$$

where the expectation is with respect to the probability measure

$$\begin{aligned} P_{0:n}^N(d(x_{0:n}^{1:N}, a_{1:n}^{1:N})) &= \prod_{i=1}^N p_n^y(dx_n^i) \prod_{\ell=2}^n \left\{ \prod_{j=1}^N \sum_{k=1}^N \omega_{\ell-1}^k \delta_k(da_\ell^j) p_{\ell-1}^y(dx_{\ell-1}^j | x_\ell^{a_\ell^j}) \right\} \\ &\quad \times \prod_{j=1}^N \sum_{k=1}^N \omega_0^k \delta_k(da_1^j) p_0^y(dx_0^j | x_1^{a_1^j}) \delta_y(d\bar{x}_0^j), \end{aligned} \quad (\text{B.3})$$

where  $\omega_t^i := \tilde{\omega}_t(\xi_{t+1}^i) / \sum_{j=1}^N \tilde{\omega}_t(\xi_{t+1}^j)$  and which corresponds to the joint law of all the random variables generated by Algorithm 1. It then follows by definition that for any  $C \in \mathcal{B}(\mathbb{R}^{\mathbf{d}_x})$ ,

$$\int \Phi_{0:n}^N(dz_{0:n}) \mathbb{1}_C(z_0) = \mathbb{E} \left[ \int \phi_{0:n}^N(dz_{0:n}) \mathbb{1}_C(z_0) \right] = \mathbb{E}[\phi_0^N(C)] = \Phi_0^N(C).$$

Define also the law of the *conditional* particle cloud

$$\begin{aligned} \mathbf{P}^N(d(x_{0:n}^{1:N}, a_{1:n}^{1:N}) | z_{0:n}) &= \delta_{z_n}(dx_n^N) \prod_{i=1}^{N-1} p_n^y(dx_n^i) \\ &\quad \times \prod_{\ell=2}^n \delta_{z_{\ell-1}}(dx_{\ell-1}^N) \delta_N(da_{\ell-1}^N) \prod_{j=1}^{N-1} \sum_{k=1}^N \omega_{\ell-1}^k \delta_k(da_\ell^j) p_{\ell-1}^y(dx_{\ell-1}^j | x_\ell^{a_\ell^j}) \\ &\quad \times \delta_{z_0}(dx_0^N) \delta_N(da_1^N) \prod_{j=1}^{N-1} \sum_{k=1}^N \omega_0^k \delta_k(da_1^j) p_0^y(dx_0^j | x_1^{a_1^j}) \delta_y(d\bar{x}_0^j). \end{aligned} \quad (\text{B.4})$$

In what follows  $\mathbb{E}_{z_{0:n}}$  refers to expectation with respect to  $\mathbf{P}^N(\cdot | z_{0:n})$ . Finally, for  $s \in [0 : n - 1]$  we let  $\Omega_s^N$  denote the sum of the filtering weights at step  $s$ , i.e.  $\Omega_s^N = \sum_{i=1}^N \tilde{\omega}_s(\xi_{s+1}^i)$ . We also write  $\mathcal{Z}_0 = \int p_0(x_0) \delta_y(d\bar{x}_0) dx_0$  and for all  $\ell \in [1 : n]$ ,  $\mathcal{Z}_\ell = \int \bar{q}_{\ell|0}(\bar{x}_\ell | y) p_\ell(dx_\ell)$ .

The proof of Proposition 2.2 relies on two Lemmata stated below and proved in Appendix B.1; in Lemma B.1 we provide an expression for the Radon-Nikodym derivative  $d\phi_{0:n}^y/d\Phi_{0:n}^y$  and in Lemma B.2 we explicit its leading term.

**Lemma B.1.**  $\phi_{0:n}^y$  and  $\Phi_{0:n}^N$  are equivalent and we have that

$$\Phi_{0:n}^N(dz_{0:n}) = \mathbb{E}_{z_{0:n}} \left[ \frac{N^n \mathcal{Z}_0 / \mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right] \phi_{0:n}^y(dz_{0:n}). \quad (\text{B.5})$$

**Lemma B.2.** It holds that

$$\begin{aligned} \frac{\mathcal{Z}_n}{\mathcal{Z}_0} \mathbb{E}_{z_{0:n}} \left[ \prod_{s=0}^{n-1} N^{-1} \Omega_s^N \right] &= \left( \frac{N-1}{N} \right)^n \\ &\quad + \frac{(N-1)^{n-1}}{N^n} \sum_{s=1}^n \frac{\mathcal{Z}_s / \mathcal{Z}_0}{\bar{q}_{s|0}(\bar{z}_s | y)} \int p_{0|s}(x_0 | z_s) \delta_y(d\bar{x}_0) dx_0 + \frac{D_{0:n}^y}{N^2}. \end{aligned} \quad (\text{B.6})$$

where  $D_{0:n}^y$  is a positive constant.

Before proceeding with the proof of Proposition 2.2, let us note that having  $z \mapsto \tilde{\omega}_\ell(z)$  bounded on  $\mathbb{R}^{\mathbf{d}_x}$  for all  $\ell \in [0 : n - 1]$  is sufficient to guarantee that  $C_{0:n}^y$  and  $D_{0:n}^y$  are finite since in this case it follows immediately that  $\mathbb{E}_{z_{0:n}} \left[ \prod_{s=0}^{n-1} N^{-1} \Omega_s^N \right]$  is bounded and so is the right hand side of (B.6). This can be achieved with a slight

modification of (2.5). Indeed, consider instead the following recursion for  $s \in [0 : n]$  where  $\delta > 0$ ,

$$\begin{aligned}\phi_n^y(x_n) &\propto (\bar{q}_{n|0}(\bar{x}_n|y) + \delta) p_n(x_n), \\ \phi_s^y(x_s) &\propto \int \phi_{s+1}^y(x_{s+1}) p_s(dx_s|x_{s+1}) \frac{\bar{q}_s(\bar{x}_s|y) + \delta}{\bar{q}_{s+1}(\bar{x}_{s+1}|y) + \delta} dx_{s+1}.\end{aligned}$$

Then we have that

$$\phi_0^y(x_0) \propto \int \phi_1^y(x_1) \underline{p}_0(\underline{x}_0|x_1) \frac{\bar{p}_0(y|x_1)}{\bar{q}_{1|0}(\bar{x}_1|y) + \delta} dx_1.$$

We can then use Algorithm 1 to produce a particle approximation of  $\phi_0^y$  using the following transition and weight function,

$$\begin{aligned}p_s^{y,\delta}(x_s|x_{s+1}) &= \frac{\gamma_s(y|x_{s+1})}{\gamma_s(y|x_{s+1}) + \delta} p_s^y(x_s|x_{s+1}) + \frac{\delta}{\gamma_s(y|x_{s+1}) + \delta} p_s(x_s|x_{s+1}), \\ \tilde{\omega}_s(x_{s+1}) &= (\gamma_s(y|x_{s+1}) + \delta) / (\bar{q}_{s+1|0}(\bar{x}_{s+1}|y) + \delta),\end{aligned}$$

where  $\gamma_s(y|x_{s+1}) = \int \bar{q}_{s|0}(\bar{x}_s|y) p_s(x_s|x_{s+1}) dx_s$  is available in closed form and  $p_s^y$  is defined in (2.3).  $\tilde{\omega}_s$  is thus clearly bounded for all  $s \in [0 : n - 1]$  and it is still possible to sample from  $p_s^{y,\delta}$  since it is simply a mixture between the transition (2.3) and the ‘‘prior’’ transition.

*Proof of Proposition 2.2.* Consider the *forward* Markov kernel

$$\vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}) = \frac{\mathbf{p}_{1:n}(dz_{1:n}) p_0(z_0|z_1)}{\int \mathbf{p}_{1:n}(d\tilde{z}_{1:n}) p_0(\tilde{z}_0|\tilde{z}_1)}, \quad (\text{B.7})$$

which satisfies

$$\phi_{0:n}^y(dz_{0:n}) = \phi_0^y(dz_0) \vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}).$$

By Lemma B.1 we have for any  $C \in \mathcal{B}(\mathbb{R}^{d_x})$  that

$$\begin{aligned}\Phi_0^N(C) &= \int \Phi_{0:n}^N(dz_{0:n}) \mathbb{1}_C(z_0) \\ &= \int \mathbb{1}_C(z_0) \mathbb{E}_{z_{0:n}} \left[ \frac{N^n \mathcal{Z}_0 / \mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right] \phi_{0:n}^y(dz_{0:n}) \\ &= \int \mathbb{1}_C(z_0) \int \vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}) \mathbb{E}_{z_{0:n}} \left[ \frac{N^n \mathcal{Z}_0 / \mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right] \phi_0^y(dz_0),\end{aligned}$$

which shows that the Radon-Nikodym derivative  $d\Phi_0^N/d\phi_0^y$  is,

$$\frac{d\Phi_0^N}{d\phi_0^y}(z_0) = \int \vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}) \mathbb{E}_{z_{0:n}} \left[ \frac{N^n \mathcal{Z}_0 / \mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right].$$

Applying Jensen’s inequality twice yields

$$\frac{d\Phi_0^N}{d\phi_0^y}(z_0) \geq \frac{N^n \mathcal{Z}_0 / \mathcal{Z}_n}{\int \vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}) \mathbb{E}_{z_{0:n}} \left[ \prod_{s=0}^{n-1} N^{-1} \Omega_s^N \right]},$$

and it then follows that

$$\mathsf{KL}(\phi_0^y \| \Phi_0^N) \leq \int \log \left( \frac{\mathcal{Z}_n}{\mathcal{Z}_0} \int \vec{\mathbf{B}}_{1:n}(z_0, dz_{1:n}) \mathbb{E}_{z_{0:n}} \left[ \prod_{s=0}^{n-1} N^{-1} \Omega_s^N \right] \right) \phi_0^y(dz_0).$$

Finally, using Lemma B.2 and the fact that  $\log(1 + x) < x$  for  $x > 0$  we get

$$\mathsf{KL}(\phi_0^y \| \Phi_0^N) \leq \frac{C_{0:n}^y}{N-1} + \frac{D_{0:n}^y}{N^2}$$

where

$$\mathcal{C}_{0:n}^y := \sum_{s=1}^n \int \frac{\mathcal{Z}_s / \mathcal{Z}_0}{\bar{q}_{s|0}(\bar{z}_s|y)} \left( p_{0|s}(x_0|z_s) \delta_y(d\bar{x}_0) d\underline{x}_0 \right) \phi_s^y(dz_s),$$

and  $\phi_s^y(z_s) \propto p_s(z_s) \int p_{0|s}(z_0|z_s) \delta_y(d\bar{z}_0) d\bar{x}_0$ .  $\square$

### Proof of Lemma B.1 and Lemma B.2

*Proof of Lemma B.1.* We have that

$$\begin{aligned} & \Phi_{0:n}^N(dz_{0:n}) \\ &= N^{-1} \int P_{0:n}^N(dx_{0:n}^{1:N}, da_{1:n}^{1:N}) \sum_{k_{0:n} \in [1:N]^{n+1}} \delta_{y \sim \underline{x}_0^{k_0}}(dz_0) \prod_{s=1}^n \mathbb{1}\{k_s = a_s^{k_{s-1}}\} \delta_{x_s^{k_s}}(dz_s) \\ &= N^{-1} \int \sum_{k_{0:n}} \sum_{a_{1:n}^{1:N}} \delta_{y \sim \underline{x}_0^{k_0}}(dz_0) \prod_{s=1}^n \mathbb{1}\{k_s = a_s^{k_{s-1}}\} \delta_{x_s^{k_s}}(dz_s) \\ &\quad \times \prod_{j=1}^N p_n^y(dx_n^j) \left\{ \prod_{\ell=2}^N \prod_{i=1}^N \omega_{\ell-1}^{a_\ell^i} p_{\ell-1}^y(dx_{\ell-1}^i | x_\ell^{a_\ell^i}) \right\} \prod_{r=1}^N \omega_r^{a_r^r} p_{\ell-1}^y(dx_0^r | x_1^{a_1^r}) \delta_y(\bar{x}_0^r) \\ &= N^{-1} \int \sum_{k_{0:n}} \sum_{a_{1:n}^{1:N}} p_n^y(dx_n^{k_n}) \delta_{x_n^{k_n}}(dz_n) \prod_{j \neq k_n} p_n^y(dx_n^j) \prod_{\ell=2}^n \left\{ \prod_{i \neq k_{\ell-1}} \omega_{\ell-1}^{a_\ell^i} p_{\ell-1}^y(dx_{\ell-1}^i | x_\ell^{a_\ell^i}) \right. \\ &\quad \times \left. \mathbb{1}\{a_\ell^{k_{\ell-1}} = k_\ell\} \frac{\tilde{\omega}_{\ell-1}(x_\ell^{a_\ell^{k_{\ell-1}}})}{\Omega_{\ell-1}^N} p_{\ell-1}^y(dx_\ell^{k_{\ell-1}} | x_\ell^{a_\ell^{k_{\ell-1}}}) \delta_{x_{\ell-1}}(dz_{\ell-1}) \right\} \\ &\quad \times \left\{ \prod_{r \neq k_0} \omega_r^{a_r^r} p_0^y(dx_0^r | x_1^{a_1^r}) \delta_y(\bar{x}_0^r) \right\} \mathbb{1}\{a_1^{k_0} = k_1\} \frac{\tilde{\omega}_0(x_1^{a_1^{k_0}})}{\Omega_0^N} p_0^y(d\underline{x}_0^{k_0} | x_0^{a_1^{k_0}}) \delta_{y \sim \underline{x}_0^{k_0}}(dz_0). \end{aligned}$$

Then, using that for all  $s \in [2 : n]$

$$\tilde{\omega}_{s-1}(x_s^{k_s}) p_{s-1}^y(dx_{s-1}^{k_{s-1}} | x_s^{k_s}) = \frac{\bar{q}_{s-1|0}(\bar{x}_{s-1}^{k_{s-1}} | y)}{\bar{q}_{s|0}(\bar{x}_s^{k_s} | y)} p_s(dx_{s-1}^{k_{s-1}} | x_s^{k_s}),$$

we recursively get that

$$\begin{aligned} & p_n^y(dx_n^{k_n}) \delta_{x_n^{k_n}}(dz_n) \prod_{s=2}^n \mathbb{1}\{a_s^{k_{s-1}} = k_s\} \frac{\tilde{\omega}_{s-1}(x_s^{k_{s-1}})}{\Omega_{s-1}^N} p_{s-1}^y(dx_{s-1}^{k_{s-1}} | x_s^{k_{s-1}}) \delta_{x_{s-1}}(dz_{s-1}) \\ &\quad \times \mathbb{1}\{a_1^{k_0} = k_1\} \frac{\tilde{\omega}_0(x_1^{k_0})}{\Omega_0^N} p_0^y(d\underline{x}_0^{k_0} | x_1^{k_0}) \delta_{y \sim \underline{x}_0^{k_0}}(dz_0) \\ &= \frac{\bar{q}_{n|0}(z_n | y) p_n(dz_n)}{\mathcal{Z}_n} \delta_{z_n}(dx_n^{k_n}) \prod_{s=2}^n \mathbb{1}\{a_s^{k_{s-1}} = k_s\} \frac{\bar{q}_{s-1|0}(\bar{z}_{s-1} | y)}{\Omega_{s-1}^N \bar{q}_{s|0}(\bar{z}_s | y)} p_{s-1}(dz_{s-1} | z_s) \delta_{z_{s-1}}(dx_{s-1}^{k_{s-1}}) \\ &\quad \times \mathbb{1}\{a_1^{k_0} = k_1\} \frac{\bar{p}_0(y | z_1)}{\Omega_0^N \bar{q}_{1|0}(\bar{z}_1 | y)} p_0(d\underline{z}_0 | z_1) \delta_y(d\bar{z}_0) \delta_{z_0}(dx_0^{k_0}) \\ &= \frac{\mathcal{Z}_0}{\mathcal{Z}_n} \phi_{0:n}^y(dz_{0:n}) \delta_{z_n}(dx_n^{k_n}) \prod_{s=1}^n \mathbb{1}\{a_s^{k_{s-1}} = k_s\} \frac{1}{\Omega_{s-1}^N} \delta_{z_{s-1}}(dx_{s-1}^{k_{s-1}}). \end{aligned}$$

Thus, we obtain

$$\begin{aligned}
\Phi_{0:n}^N(dz_{0:n}) &= N^{-1} \int \sum_{k_{0:n}} \sum_{a_{1:n}^{1:N}} \phi_{0:n}^y(dz_{0:n}) \frac{\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \delta_{z_n}(dx_n^{k_n}) \prod_{j \neq k_n} p_n^y(dx_n^j) \\
&\quad \times \prod_{\ell=2}^n \mathbb{1}\{a_\ell^{k_{\ell-1}} = k_\ell\} \delta_{z_{\ell-1}}(dx_{\ell-1}^{k_{\ell-1}}) \prod_{i \neq k_{\ell-1}} \omega_{\ell-1}^{a_i^i} p_{\ell-1}^y(dx_{\ell-1}^i | x_\ell^{a_\ell^i}) \\
&\quad \times \mathbb{1}\{a_1^{k_0} = k_1\} \delta_{z_0}(dx_0^{k_0}) \prod_{i \neq k_0} \omega_0^{a_1^i} p_0^y(\underline{x}_0^i | x_1^{a_1^i}) \delta_y(d\bar{x}_0) \\
&= N^{-1} \sum_{k_{0:n}} \phi_{0:n}^y(dz_{0:n}) \mathbb{E}_{z_{0:n}}^{k_{0:n}} \left[ \frac{\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right],
\end{aligned}$$

where for all  $k_{0:n} \in [1 : N]^{n+1}$   $\mathbb{E}_{z_{0:n}}^{k_{0:n}}$  denotes the expectation under the Markov kernel

$$\begin{aligned}
\mathbf{P}_{k_{0:n}}^N(dx_{0:n}^{1:N}, a_{1:n}^{1:N})|z_{0:n}) &= \delta_{z_n}(dx_n^{k_n}) \prod_{i \neq k_n} p_n^y(dx_n^i) \\
&\quad \times \prod_{\ell=2}^n \delta_{z_{\ell-1}}(dx_{\ell-1}^{k_{\ell-1}}) \delta_{k_\ell}(da_\ell^{k_{\ell-1}}) \prod_{j \neq k_{\ell-1}} \sum_{k=1}^N \omega_{\ell-1}^k \delta_k(da_\ell^j) p_{\ell-1}^y(dx_{\ell-1}^j | x_\ell^{a_\ell^j}) \\
&\quad \times \delta_{z_0}(dx_0^{k_0}) \delta_{k_1}(da_1^{k_0}) \prod_{j \neq k_0} \sum_{k=1}^N \omega_0^k \delta_k(da_1^j) p_0^y(\underline{x}_0^j | x_1^{a_1^j}) \delta_y(d\bar{x}_0).
\end{aligned}$$

Note however that for all  $(k_{0:n}, \ell_{0:n}) \in ([1 : N]^{n+1})^2$ ,

$$\mathbb{E}_{z_{0:n}}^{k_{0:n}} \left[ \frac{1}{\prod_{s=0}^{n-1} \Omega_s^N} \right] = \mathbb{E}_{z_{0:n}}^{\ell_{0:n}} \left[ \frac{1}{\prod_{s=0}^{n-1} \Omega_s^N} \right]$$

and thus it follows that

$$\Phi_{0:n}^N(dz_{0:n}) = \mathbb{E}_{z_{0:n}} \left[ \frac{N^n \mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right] \phi_{0:n}^y(dz_{0:n}). \quad (\text{B.8})$$

□

Denote by  $\{\mathcal{F}_s\}_{s=0}^n$  the filtration generated by a conditional particle cloud sampled from the kernel  $\mathbf{P}^N$  (B.4), i.e. for all  $\ell \in [0 : n - 1]$

$$\mathcal{F}_s = \sigma(\xi_{s:n}^{1:N}, I_{s+1:n}^{1:N}).$$

and  $\mathcal{F}_n = \sigma(\xi_n^{1:N})$ . Define for all bounded  $f$  and  $\ell \in [0 : n - 1]$

$$\gamma_{\ell:n}^N(f) = \left\{ \prod_{s=\ell+1}^{n-1} N^{-1} \Omega_s^N \right\} N^{-1} \sum_{k=1}^N \tilde{\omega}_\ell(\xi_{\ell+1}^k) f(\xi_{\ell+1}^k), \quad (\text{B.9})$$

with the convention  $\gamma_{\ell:n}^N(f) = 1$  if  $\ell \geq n$ . Define also the transition Kernel

$$Q_{\ell-1|\ell+1}^y : \mathbb{R}^{\mathbf{d}_x} \times \mathcal{B}(\mathbb{R}^{\mathbf{d}_x}) \ni (x_{\ell+1}, A) \mapsto \int \mathbb{1}_A(x_\ell) \tilde{\omega}_{\ell-1}(x_\ell) p_\ell^y(dx_\ell | x_{\ell+1}). \quad (\text{B.10})$$

Using eqs. (2.3) and (2.4), it is easily seen that for all  $\ell \in [0 : n - 1]$ ,

$$\tilde{\omega}_\ell(x_{\ell+1}) Q_{\ell-1|\ell+1}^y(f)(x_{\ell+1}) = \frac{1}{\bar{q}_{\ell+1|0}(\bar{x}_{\ell+1}|y)} \int \bar{q}_{\ell|0}(\bar{x}_s|y) \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell^y(dx_\ell | x_{\ell+1}). \quad (\text{B.11})$$

Define  $\mathbf{1} : x \in \mathbb{R}^{\mathbf{d}_x} \mapsto 1$ . We may thus write that  $\gamma_{\ell:n}^N(f) = N^{-1} \gamma_{\ell+1:n}^N(\mathbf{1}) \sum_{k=1}^N \tilde{\omega}_\ell(\xi_{\ell+1}^k) f(\xi_{\ell+1}^k)$ .

**Lemma B.3.** For all  $\ell \in [0 : n - 1]$  it holds that

$$\mathbb{E}_{z_{0:n}} [\gamma_{\ell-1:n}^N(f)] = \frac{N-1}{N} \mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N(Q_{\ell-1|\ell+1}^y(f))] + \frac{1}{N} \mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N(\mathbf{1})] \tilde{\omega}_{\ell-1}(z_\ell) f(z_\ell).$$

*Proof.* By the tower property and the fact that  $\gamma_{\ell:n}^N(f)$  is  $\mathcal{F}_{\ell+1}$ -measurable, we have that

$$\mathbb{E}_{z_{0:n}} [\gamma_{\ell-1:n}^N(f)] = \mathbb{E}_{z_{0:n}} \left[ N^{-1} \gamma_{\ell+1:n}^N(\mathbf{1}) \Omega_\ell^N \mathbb{E}_{z_{0:n}} \left[ N^{-1} \sum_{k=1}^N \tilde{\omega}_{\ell-1}(\xi_\ell^k) f(\xi_\ell^k) \middle| \mathcal{F}_{\ell+1} \right] \right].$$

Note that for all  $\ell \in [0 : n - 1]$ ,  $(\xi_\ell^1, \dots, \xi_\ell^{N-1})$  are identically distributed conditionally on  $\mathcal{F}_{\ell+1}$  and

$$\mathbb{E}_{z_{0:n}} \left[ \tilde{\omega}_{\ell-1}(\xi_\ell^j) f(\xi_\ell^j) \middle| \mathcal{F}_{\ell+1} \right] = \frac{1}{\Omega_\ell^N} \sum_{k=1}^N \tilde{\omega}_\ell(\xi_{\ell+1}^k) \int \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell^y(dx_\ell | \xi_{\ell+1}^k),$$

leading to

$$\begin{aligned} \mathbb{E}_{z_{0:n}} \left[ N^{-1} \sum_{k=1}^N \tilde{\omega}_{\ell-1}(\xi_\ell^k) f(\xi_\ell^k) \middle| \mathcal{F}_{\ell+1} \right] \\ = \frac{N-1}{N \Omega_\ell^N} \sum_{k=1}^N \tilde{\omega}_\ell(\xi_{\ell+1}^k) \int \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell^y(dx_\ell | \xi_{\ell+1}^k) + \frac{1}{N} \tilde{\omega}_{\ell-1}(z_\ell) f(z_\ell), \end{aligned}$$

and the desired recursion follows.  $\square$

*Proof of Lemma B.2.* We proceed by induction and show for all  $\ell \in [0 : n - 2]$

$$\begin{aligned} & \mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N(f)] \\ &= \left( \frac{N-1}{N} \right)^{n-\ell} \frac{\int p_{\ell+1}(dx_{\ell+1}) \bar{q}_{\ell+1|0}(\bar{x}_{\ell+1}|y) \tilde{\omega}_\ell(x_{\ell+1}) f(x_{\ell+1})}{\mathcal{Z}_n} \\ &+ \frac{(N-1)^{n-\ell-1}}{N^{n-\ell}} \left[ (\mathcal{Z}_{\ell+1}/\mathcal{Z}_n) f(z_{\ell+1}) \tilde{\omega}_\ell(z_{\ell+1}) \right. \\ &+ \left. \sum_{s=\ell+2}^n \frac{\mathcal{Z}_s/\mathcal{Z}_n}{\bar{q}_{s|0}(\bar{z}_s|y)} \int \tilde{\omega}_\ell(x_{\ell+1}) \bar{q}_{\ell+1|0}(\bar{x}_{\ell+1}|y) f(x_{\ell+1}) p_{\ell+1|s}(dx_{\ell+1}|z_s) \right] + \frac{D_{\ell:n}^y}{N^2}. \end{aligned} \tag{B.12}$$

where  $f$  is a bounded function and  $D_{\ell:n}^y$  is a positive constant. The desired result in Lemma B.2 then follows by taking  $\ell = 0$  and  $f = \mathbf{1}$ .

Assume that (B.12) holds at step  $\ell$ . To show that it holds at step  $\ell - 1$  we use Lemma B.3 and we compute  $\mathbb{E}_{z_{0:n}} [\gamma_{\ell-1:n}^N(Q_{\ell-1|\ell+1}^y(f))]$  and  $\mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N(\mathbf{1})] \tilde{\omega}_{\ell-1}(z_\ell) f(z_\ell)$ .

Using the following identities which follow from (B.11)

$$\begin{aligned} & \int \bar{q}_{\ell+1|0}(\bar{x}_{\ell+1}|y) \tilde{\omega}_\ell(x_{\ell+1}) Q_{\ell-1|\ell+1}^y(f)(x_{\ell+1}) p_{\ell+1}(dx_{\ell+1}) \\ &= \int \bar{q}_{\ell|0}(\bar{x}_\ell|y) \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell(dx_\ell), \end{aligned}$$

and

$$\begin{aligned} & \int \tilde{\omega}_\ell(x_{\ell+1}) \bar{q}_{\ell+1|0}(\bar{x}_{\ell+1}|y) Q_{\ell-1|\ell+1}^y(f)(x_{\ell+1}) p_{\ell+1|s}(dx_{\ell+1}|x_s) \\ &= \int \tilde{\omega}_{\ell-1}(x_\ell) \bar{q}_{\ell|0}(\bar{x}_\ell|y) f(x_\ell) p_{\ell|s}(dx_\ell|x_s), \end{aligned}$$

we get by (B.12) that

$$\begin{aligned}
& \frac{N-1}{N} \mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N (Q_{\ell-1|\ell+1}^y(f))] \\
&= \left( \frac{N-1}{N} \right)^{n-\ell+1} \frac{\int \bar{q}_{\ell|0}(\bar{x}_\ell|y) \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell(dx_\ell)}{\mathcal{Z}_n} \\
&\quad + \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}} \left[ \frac{\mathcal{Z}_{\ell+1}/\mathcal{Z}_n}{\bar{q}_{\ell+1|0}(\bar{z}_{\ell+1}|y)} \int \bar{q}_{\ell|0}(\bar{x}_\ell|y) \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell(dx_\ell|z_{\ell+1}) \right. \\
&\quad \left. + \sum_{s=\ell+2}^n \frac{\mathcal{Z}_s/\mathcal{Z}_n}{\bar{q}_{s|0}(\bar{z}_s|y)} \int \tilde{\omega}_{\ell-1}(x_\ell) \bar{q}_{\ell|0}(\bar{x}_\ell|y) f(x_\ell) p_{\ell|s}(dx_\ell|z_s) \right] + \frac{D_{\ell:n}^y}{N^2} \\
&= \left( \frac{N-1}{N} \right)^{n-\ell+1} \frac{\int \bar{q}_{\ell|0}(\bar{x}_\ell|y) \tilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell(dx_\ell)}{\mathcal{Z}_n} \\
&\quad + \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}} \sum_{s=\ell+1}^n \frac{\mathcal{Z}_s/\mathcal{Z}_n}{\bar{q}_{s|0}(\bar{z}_s|y)} \int \tilde{\omega}_{\ell-1}(x_\ell) \bar{q}_{\ell|0}(\bar{x}_s|y) f(x_\ell) p_{\ell|s}(dx_\ell|z_s) + \frac{D_{\ell:n}^y}{N^2}. \tag{B.13}
\end{aligned}$$

The induction step is finished by using again (B.12) and noting that

$$\frac{1}{N} \mathbb{E}_{z_{0:n}} [\gamma_{\ell:n}^N(\mathbf{1})] \tilde{\omega}_{\ell-1}(z_\ell) f(z_\ell) = \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}} (\mathcal{Z}_\ell/\mathcal{Z}_n) \tilde{\omega}_{\ell-1}(z_\ell) f(z_\ell) + \frac{\tilde{D}_{\ell:n}^y}{N^2}.$$

and then setting  $D_{\ell-1:n}^y = D_{\ell:n}^y + \tilde{D}_{\ell:n}^y$ .

It remains to compute the initial value at  $\ell = n - 2$ . Note that

$$\mathbb{E}_{z_{0:n}} [\gamma_{n-1:n}^N(f)] = \frac{N-1}{N} \int p_n^y(dx_n) \tilde{\omega}_{n-1}(x_n) f(x_n) + \frac{1}{N} \tilde{\omega}_{n-1}(z_n) f(z_n) \tag{B.14}$$

and thus by Lemma B.3 and similarly to the previous computations

$$\begin{aligned}
& \mathbb{E}_{z_{0:n}} [\gamma_{n-2:n}^N(f)] \\
&= \left( \frac{N-1}{N} \right)^2 \int p_n^y(dx_n) \tilde{\omega}_{n-1}(x_n) Q_{n-2|n}^y(f)(x_n) + \frac{N-1}{N^2} \left[ \tilde{\omega}_{n-1}(z_n) Q_{n-2|n}^y(f)(z_n) \right. \\
&\quad \left. + \tilde{\omega}_{n-2}(z_{n-1}) f(z_{n-1}) \int p_n^y(dx_n) \tilde{\omega}_{n-1|n}(x_n) \right] + \frac{D_{n-2fa:n}^y}{N^2} \\
&= \left( \frac{N-1}{N} \right)^2 \frac{\int \bar{q}_{n-1|0}(x_{n-1}|y) \tilde{\omega}_{n-2}(x_{n-1}) p_{n-1}(dx_{n-1})}{\mathcal{Z}_n} \\
&\quad + \frac{N-1}{N^2} \left[ (\mathcal{Z}_{n-1}/\mathcal{Z}_n) \tilde{\omega}_{n-2}(z_{n-1}) f(z_{n-1}) \right. \\
&\quad \left. + \frac{1}{\bar{q}_{n|0}(\bar{x}_n|y)} \int \bar{q}_{n-1|0}(\bar{x}_{n-1}|y) \tilde{\omega}_{n-2}(x_{n-1}) f(x_{n-1}) p_{n-1}(dx_{n-1}|z_n) \right] + \frac{D_{n-2:n}^y}{N^2}.
\end{aligned}$$

□

## B.2 Proof of Proposition 2.3

In this section and only in this section we make the following assumption

**(A2)** For all  $s \in [0 : n - 1]$ ,  $p_s(x_s) q_{s+1}(x_{s+1}|x_s) = p_{s+1}(x_{s+1}) \lambda_s(x_s|x_{s+1})$ .

We also consider  $\sigma_\delta = 0$ . In what follows we let  $\tau_{d_y+1} = n$  and we write  $\tau_{1:d_y} = \{\tau_1, \dots, \tau_{d_y}\}$  and  $\overline{\tau_{1:d_y}} = [1:n] \setminus \tau_{1:t}$ . Define the measure

$$\Gamma_{0:n}^y(dx_{0:n}) = p_n(dx_n) \prod_{s \in \overline{\tau_{1:d_y}}} \lambda_s(dx_s | x_{s+1}) \prod_{i=1}^{d_y} \lambda_{\tau_i}(x_{\tau_i} | x_{\tau_i+1}) dx_{\tau_i}^{\backslash i} \delta_{y[i]}(dx_{\tau_i}[i]). \quad (\text{B.15})$$

Under (A2) it has the following alternative *forward* expression,

$$\Gamma_{0:n}^y(dx_{0:n}) = p_0(dx_0) \prod_{s \in \overline{\tau_{1:d_y}}} q_{s+1}(dx_{s+1} | x_s) \prod_{i=1}^{d_y} q_{\tau_i}(x_{\tau_i} | x_{\tau_i-1}) dx_{\tau_i}^{\backslash i} \delta_{y[i]}(dx_{\tau_i}[i]). \quad (\text{B.16})$$

Since the forward kernels decompose over the dimensions of the states, i.e.

$$q_{s+1}(x_{s+1} | x_s) = \prod_{\ell=1}^{d_x} q_{s+1}^\ell(x_{s+1}[\ell] | x_s[\ell])$$

where  $q_{s+1}^\ell(x_{s+1}[\ell] | x_s[\ell]) = \mathcal{N}(x_{s+1}[\ell]; (\alpha_{s+1}/\alpha_s)^{1/2} x_s[\ell], 1 - (\alpha_{s+1}/\alpha_s))$ , we can write

$$\Gamma_{0:n}^y(x_{0:n}) = p_0(x_0) \prod_{\ell=1}^{d_x} \Gamma_{1:n|0,\ell}^y(x_1[\ell], \dots, x_n[\ell] | x_0[\ell]), \quad (\text{B.17})$$

where for  $\ell \in [1 : d_y]$

$$\Gamma_{1:n|0,\ell}^y(x_1[\ell], \dots, x_n[\ell] | x_0[\ell]) = q_{\tau_\ell}^\ell(y[\ell] | x_{\tau_\ell-1}[\ell]) \prod_{s \neq \tau_\ell} q_s^\ell(dx_s[\ell] | x_{s-1}[\ell]), \quad (\text{B.18})$$

and for  $\ell \in [d_y + 1 : d_x]$ ,

$$\Gamma_{1:n|0,\ell}^y(x_1[\ell], \dots, x_n[\ell] | x_0[\ell]) = \prod_{s=0}^{n-1} q_{s+1}^\ell(x_{s+1}[\ell] | x_s[\ell]). \quad (\text{B.19})$$

With these quantities in hand we can now prove Proposition 2.3.

*Proof of Proposition 2.3.* Note that for  $\ell \in [1 : d_y]$ ,

$$\begin{aligned} \mathcal{N}(y[\ell]; \alpha_{\tau_\ell} x_0[\ell], 1 - \alpha_{\tau_\ell}) &= q_{\tau_\ell|0}^\ell(y[\ell] | x_0[\ell]) = \int q_{\tau_\ell}^\ell(y[\ell] | x_{\tau_\ell-1}[\ell]) \prod_{s \neq \tau_\ell} q_s^\ell(dx_s[\ell] | x_{s-1}[\ell]) \\ &= \int \Gamma_{1:n|0,\ell}^y(d(x_1[\ell], \dots, x_n[\ell]) | x_0[\ell]) \end{aligned}$$

and thus by (2.11) we have that

$$\begin{aligned} p_0(x_0) g_0^y(x_0) &\propto p_0(x_0) \prod_{\ell=1}^{d_y} \mathcal{N}(y[\ell]; \alpha_{\tau_\ell} x_0[\ell], 1 - \alpha_{\tau_\ell}) \\ &= p_0(x_0) \prod_{\ell=1}^{d_y} \int \Gamma_{1:n|0,\ell}^y(d(x_1[\ell], \dots, x_n[\ell]) | x_0[\ell]) \\ &= p_0(x_0) \prod_{\ell=1}^{d_x} \int \Gamma_{1:n|0,\ell}^y(d(x_1[\ell], \dots, x_n[\ell]) | x_0[\ell]). \end{aligned}$$

By (B.16) it follows that

$$\phi_0^y(x_0) = \frac{1}{\int \Gamma_{0:n}^y(\tilde{x}_{0:n}) d\tilde{x}_{0:n}} \int \Gamma_{0:n}^y(x_{0:n}) dx_{1:n},$$

and hence by (B.16) and (B.15) we get

$$\phi_0^y(x_0) \propto \int p_{\tau_{d_y}}(x_{\tau_{d_y}}) \delta_{y[d_y]}(dx_{\tau_{d_y}}[d_y]) dx_{\tau_{d_y}}^{d_y} \left\{ \prod_{i=1}^{d_y-1} \lambda_{\tau_i|\tau_{i+1}}(x_{\tau_i}|x_{\tau_{i+1}}) \delta_{y[i]}(dx_{\tau_i}[i]) dx_{\tau_i}^{d_y} \right\} \lambda_{0|\tau_1}(x_0|x_{\tau_1}).$$

This completes the proof.  $\square$

### B.3 Algorithmic details and numerics

#### B.3.1 GMM

For a given dimension  $d_x$ , we consider  $q_{\text{data}}$  a mixture of 25 Gaussian random variables. The Gaussian random variables have mean  $\mu_{i,j} := (8i, 8j, \dots, 8i, 8j) \in \mathbb{R}^{d_x}$  for  $(i, j) \in \{-2, -1, 0, 1, 2\}^2$  and unit variance. The mixture (unnormalized) weights  $\omega_{i,j}$  are independently drawn according to a  $\chi^2$  distribution. The  $\kappa$  parameter of MCGdiff is  $\kappa^2 = 10^{-4}$ . We use 20 steps of DDIM for the numerical examples and for all algorithms.

**Score:** Note that  $q_s(x_s) = \int q_{s|0}(x_s|x_0)q_{\text{data}}(x_0)dx_0$ . As  $q_{\text{data}}$  is a mixture of Gaussians,  $q_s(x_s)$  is also a mixture of Gaussians with means  $\alpha_s^{1/2} \mu_{i,j}$  and unitary variances. Therefore, using automatic differentiation libraries, we can calculate  $\nabla \log q_s(x_s)$ . Setting  $e(x_s, s) = -(1 - \alpha_s)^{1/2} \nabla \log q_s(x_s)$  leads to the optimum of (1.4).

**Forward process scaling:** We chose the sequence of  $\{\beta_s\}_{s=1}^{1000}$  as a linearly decreasing sequence between  $\beta_1 = 0.2$  and  $\beta_{1000} = 10^{-4}$ .

**Measurement model:** For a pair of dimensions  $(d_x, d_y)$  the measurement model  $(y, A, \sigma_y)$  is drawn as follows:

- $A$ : We first draw  $\tilde{A} \sim \mathcal{N}(0_{d_y \times d_x}, I_{d_y \times d_x})$  and compute the SVD decomposition of  $\tilde{A} = USV^T$ . Then, we sample for  $(i, j) \in \{-2, -1, 0, 1, 2\}^2$ ,  $s_{i,j}$  according to a uniform in  $[0, 1]$ . Finally, we set  $A = U \text{Diag}(\{s_{i,j}\}_{(i,j) \in \{-2, -1, 0, 1, 2\}^2}) V^T$ .
- $\sigma_y$ : We draw  $\sigma_y$  uniformly in the interval  $[0, \max(s_1, \dots, s_{d_y})]$ .
- $y$ : We then draw  $x_* \sim q_{\text{data}}$  and set  $y := Ax_* + \sigma_y \epsilon$  where  $\epsilon \sim \mathcal{N}(0_{d_y}, I_{d_y})$ .

**Posterior:** Once we have drawn both  $q_{\text{data}}$  and  $(y, A, \sigma_y)$ , the posterior can be exactly calculated using Bayes formula and gives a mixture of Gaussians with mixture components  $c_{i,j}$  and associated weights  $\tilde{\omega}_{i,j}$

$$c_{i,j} := \mathcal{N}(\Sigma(A^T y / \sigma_y^2 + \mu_{i,j}), \Sigma), \\ \tilde{\omega}_i := \omega_i \mathcal{N}(y; A\mu_{i,j}, \sigma^2 I_{d_x} + AA^T),$$

where  $\Sigma := (I_{d_x} + \sigma_y^{-2} A^T A)^{-1}$ .

**Variational Inference:** The RNVP entries in the numerical examination are obtained by Variational Inference using the RNVP architecture for the normalizing flow from [Dinh et al., 2017]. Given a normalizing flow  $f_\phi$  with  $\phi \in \mathbb{R}^j$ ,  $j \in \mathbb{N}_*$ , the training procedure consists of optimizing the ELBO, i.e., solving the optimization problem

$$\phi_* = \underset{\phi \in \mathbb{R}^j}{\operatorname{argmax}} \sum_{k=1}^{N_{nf}} \log |Jf_\phi(\epsilon_i)| + \log \pi_*(f_\phi(\epsilon_i)), \quad (\text{B.20})$$

where  $N_{nf} \in \mathbb{N}_*$  is the minibatch-size,  $Jf_\phi$  the Jacobian of  $f_\phi$  w.r.t  $\phi$ , and  $\epsilon_{1:N_{nf}} \sim \mathcal{N}(0, I)^{\otimes N_{nf}}$ . All the experiments were performed using a 10 layers RNVP. Equation (B.20) is solved using Adam algorithm [Kingma and Ba, 2015] with a learning rate of  $10^{-3}$  and 200 iterations with  $N_{nf} = 10$ . The losses for each pair  $(d_x, d_y)$  is shown in fig. 7, where one can see that the majority of the losses have converged.

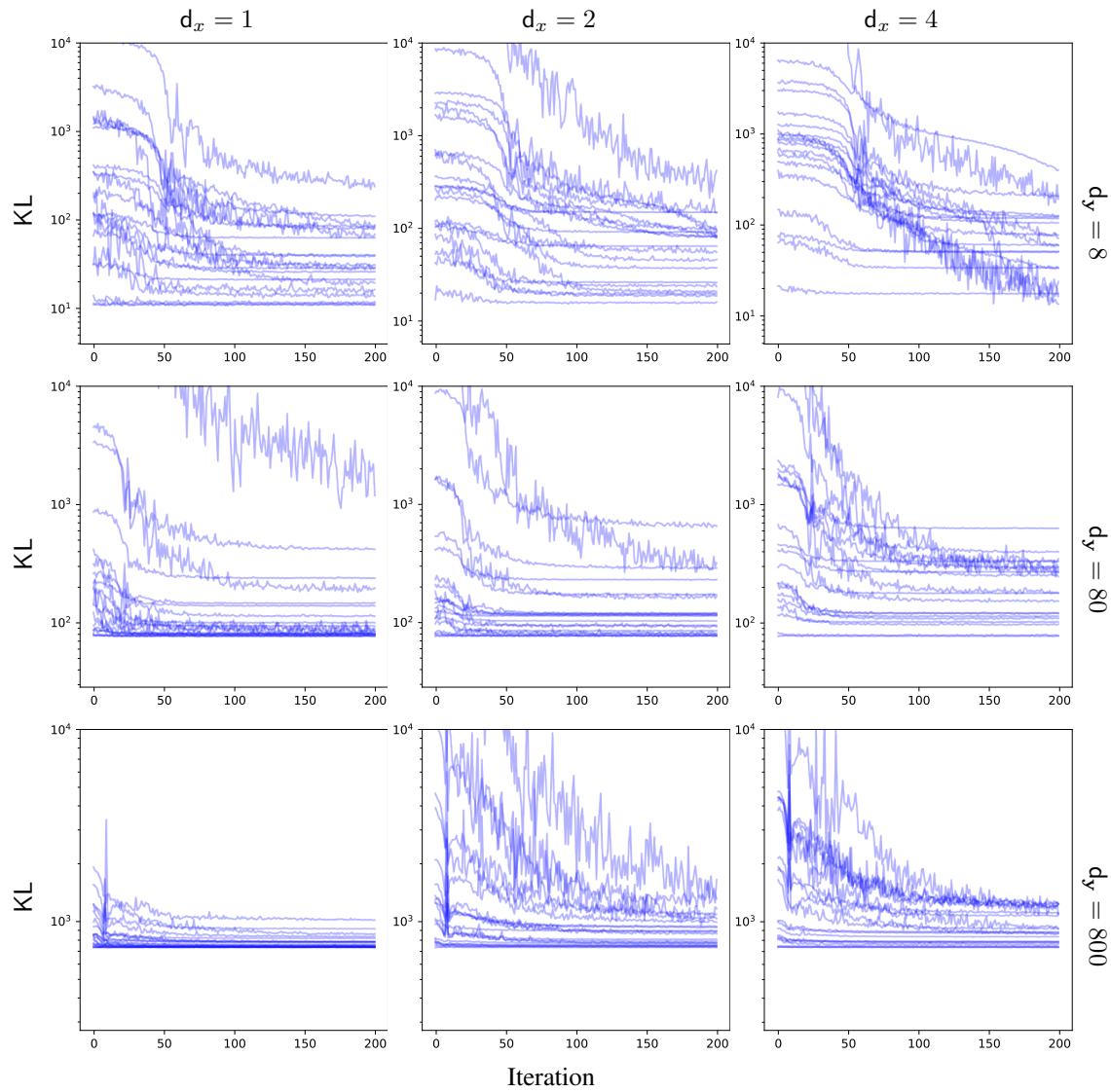


Figure 7: Evolution of KL with the number of iterations for all pairs of  $(d_x, d_y)$  tested in the GMM case.

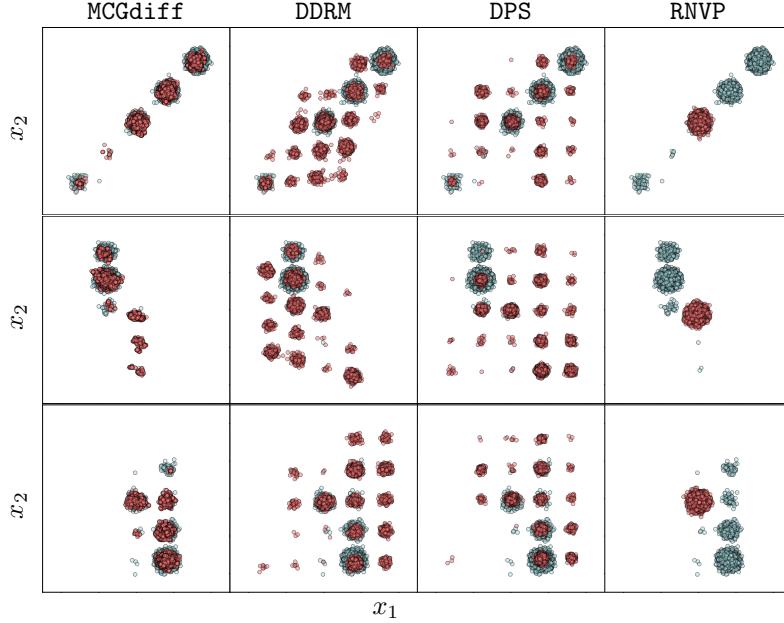


Figure 8: First two dimensions for the GMM case with  $d_x = 8$ . The rows represent  $d_y = 1, 2, 4$  respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).

**Choosing DDIM timesteps for a given measurement model:** Given a number of DDIM samples  $R$ , we choose the timesteps  $1 = t_1 < \dots < t_R = 1000 \in [1 : 1000]$  as to try to satisfy the two following constraints:

- For all  $i \in [1 : d_y]$  there exists a  $t_j$  such that  $\sigma_y \alpha_{t_j}^{1/2} \approx (1 - \alpha_{t_j})^{1/2} s_i$ ,
- For all  $i \in [1 : R - 1]$ ,  $\alpha_{t_i}^{1/2} - \alpha_{t_{i+1}}^{1/2} \approx \delta$  for some  $\delta > 0$ .

The first constraint comes naturally from the definition of  $\tau_i$ . Since the potentials have mean  $\alpha_{t_i}^{1/2} y$ , the second condition constrains the intermediate laws remain “close”. An algorithm that approximately satisfies both constraints is given below.

---

**Algorithm 2:** Timesteps choice

---

**Input:** Number of DDIM steps  $R$ ,  $\sigma_y$ ,  $\{s_i\}_{i=1}^{d_y}$ ,  $\{\alpha_i\}_{i=1}^{1000}$   
**Output:**  $\{t_j\}_{j=1}^R$   
Set  $S_\tau = \{\}$ .  
**for**  $j \leftarrow [1 : d_y]$  **do**  
  Set  $\tilde{\tau}_j = \operatorname{argmin}_{\ell \in [1:1000]} |\sigma_y \alpha_\ell^{1/2} - (1 - \alpha_\ell)^{1/2} s_j|$ .  
  Add  $\tilde{\tau}_j$  to  $S_\tau$  if  $\tilde{\tau}_j \notin S_\tau$ .  
Set  $n_m = R - \#S_\tau - 1$  and  $\delta = (\alpha_1^{1/2} - \alpha_{1000}^{1/2})/n_m$ .  
Set  $t_1 = 1$ ,  $e = 1$  and  $i_e = 1$ . **for**  $\ell \leftarrow [2 : 1000]$  **do**  
  **if**  $\alpha_e^{1/2} - \alpha_\ell^{1/2} > \delta$  **or**  $\ell \in S_\tau$  **then**  
    Set  $e = \ell$ ,  $i_e = i_e + 1$  and  $\tau_{i_e} = \ell$ .  
Set  $\tau_R = 1000$ .

---

**Additional numerics:** We now proceed to illustrate in Figures 8 to 10 the first 2 components for one of the measurement models for all the different combinations of  $(d_x, d_y)$  combinations used in table 1. We also show in fig. 11 the evolution of each observed coordinate in the noise case with  $d_y = 4$ . We can see that it follows closely the forward path of the diffused observations indicated by the blue line.

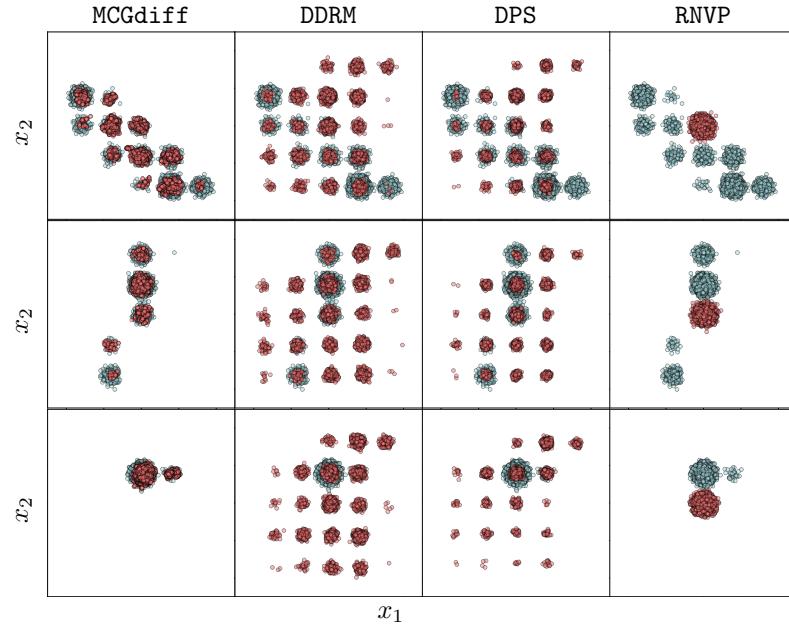


Figure 9: First two dimensions for the GMM case with  $d_x = 80$ . The rows represent  $d_y = 1, 2, 4$  respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).

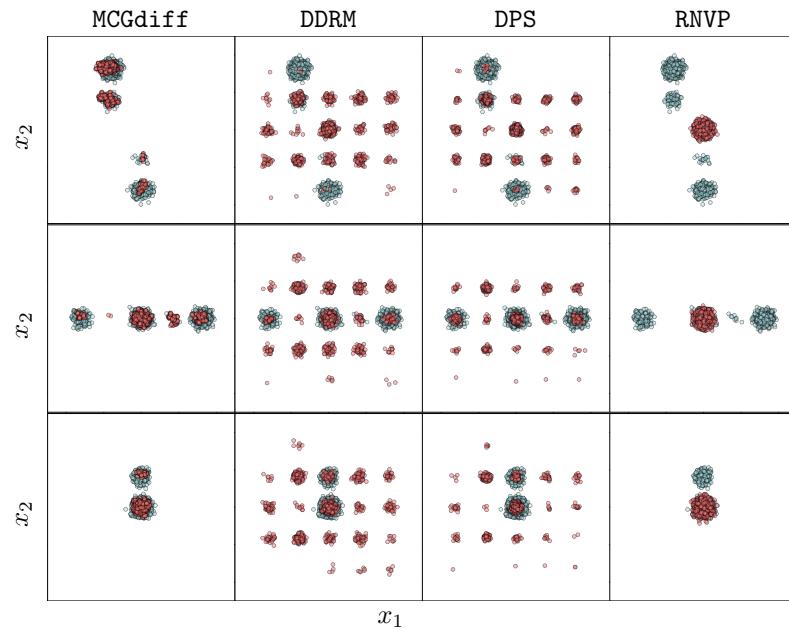


Figure 10: First two dimensions for the GMM case with  $d_x = 800$ . The rows represent  $d_y = 1, 2, 4$  respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).

$d$	$d_y$	MCGdiff	DDRM	DPS	RNVP
8	1	<b><math>1.43 \pm 0.55</math></b>	$5.88 \pm 1.16$	$4.86 \pm 1.01$	$9.43 \pm 0.99$
8	2	<b><math>0.49 \pm 0.24</math></b>	$5.20 \pm 1.32$	$5.79 \pm 1.96$	$8.93 \pm 1.29$
8	4	<b><math>0.38 \pm 0.25</math></b>	$2.51 \pm 1.29$	$3.48 \pm 1.52$	$6.71 \pm 1.54$
80	1	<b><math>1.39 \pm 0.45</math></b>	$5.64 \pm 1.10$	$4.98 \pm 1.14$	$6.86 \pm 0.88$
80	2	<b><math>0.67 \pm 0.24</math></b>	$7.07 \pm 1.35$	$5.10 \pm 1.23$	$7.79 \pm 1.50$
80	4	<b><math>0.28 \pm 0.14</math></b>	$7.81 \pm 1.48$	$4.28 \pm 1.26$	$7.95 \pm 1.61$
800	1	<b><math>2.40 \pm 1.00</math></b>	$7.44 \pm 1.15$	$6.49 \pm 1.16$	$7.74 \pm 1.34$
800	2	<b><math>1.31 \pm 0.60</math></b>	$8.95 \pm 1.12$	$6.88 \pm 1.01$	$8.75 \pm 1.02$
800	4	<b><math>0.47 \pm 0.19</math></b>	$8.39 \pm 1.48$	$5.51 \pm 1.18$	$7.81 \pm 1.63$

Table 3: Extended GMM sliced wasserstein table.

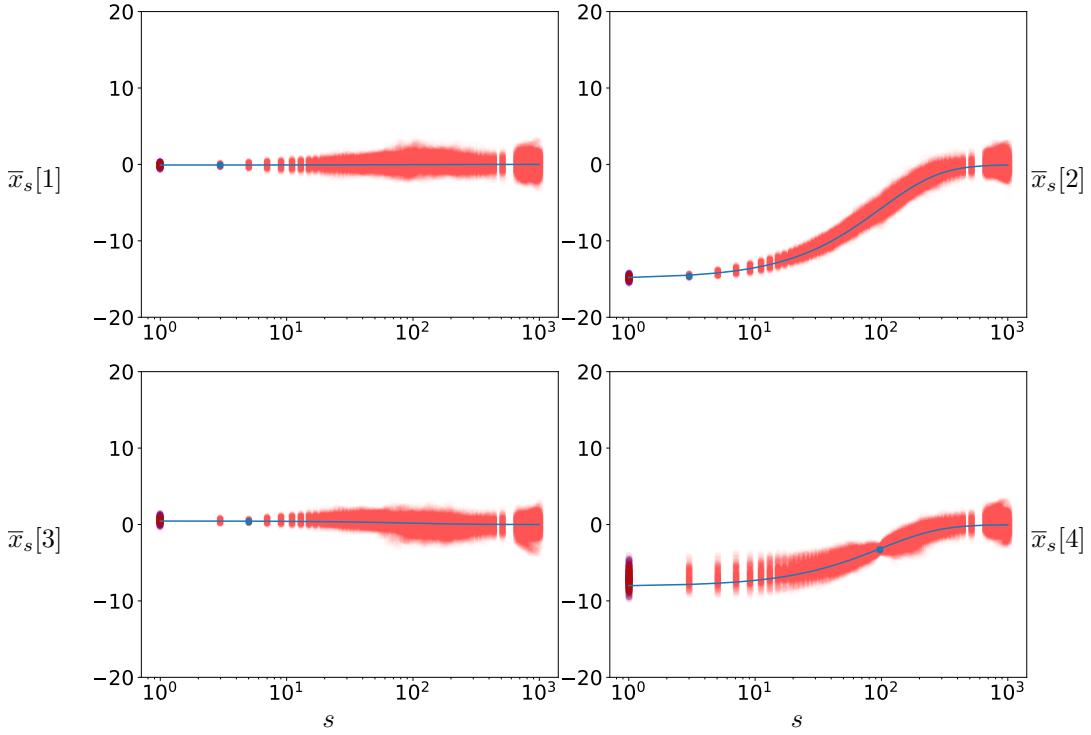


Figure 11: Illustration of the particle cloud of the 4 first observed coordinate in the case  $(d_y, d_x) = (4, 800)$  with 100 DDIM steps. The red points represent the particle cloud, while the purple points at the origin represent the posterior distribution. The blue curve corresponds to the curve  $s \rightarrow \alpha_s^{1/2} \mathbf{y}[\ell]$  and the blue dot on the curve to  $\alpha_{\tau_\ell}^{1/2} \mathbf{y}[\ell]$ .

Table 3 is an extended version of table 1.

### B.3.2 FMM

A funnel distribution is defined by the following density

$$\mathcal{N}(x_1; 0, 1) \prod_{i=1}^d \mathcal{N}(x_i; 0, \exp(x_1/2)).$$

To generate a Funnel mixture model of 20 components in dimension  $d$ , we start by firstly sampling  $(\mu_i, R_i)_{i=1}^{20}$  uniformly in  $([-20, 20]^d \times \text{SO}(R^d))^{\times 20}$ . The mixture will consist of 20 Funnel random variables translated by  $\mu_i$  and rotated by  $R_i$ , with unnormalized weights  $\omega_{i,j}$  that are independently drawn uniformly in  $[0, 1]$ .

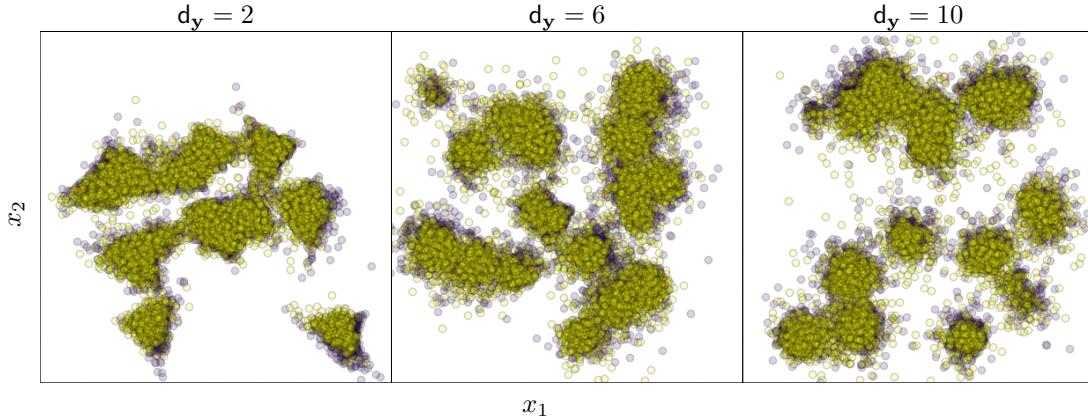


Figure 12: Purple points are samples from the prior and yellow samples from the diffusion with 25 DDIM steps.

$d$	SW
2	$0.79 \pm 0.15$
6	$0.87 \pm 0.07$
10	$0.96 \pm 0.06$

Table 4: Sliced Wasserstein between learned diffusion and target prior.

**Score** The denoising diffusion network  $e(\theta)$  in dimension  $d$  is defined as a 5 layers Resnet network where each Resnet block consists of the chaining of three blocks where each block has the following layers:

- Linear (512, 1024),
- 1d Batch Norm,
- ReLU activation.

The Resnet is preceeded by an input embedding from dimension  $d$  to 512 and in the end an output embedding layer projects the output of the resnet from 512 to  $d$ . The time  $t$  is embedded using positional embedding into dimension 512 and is added to the input at each Resnet block. The network is trained using the same loss as in [Ho et al., 2020] for  $10^4$  iterations using a batch size of 512 samples. A learning rate of  $10^{-3}$  is used for the Adam optimizer [Kingma and Ba, 2015]. Figure 12 illustrate the outcome of the learned diffusion generative model and the target prior. In table 4 we show the CLT 95% intervals for the SW between the learned diffusion generative model and the target prior.

**Forward process scaling** We chose the sequence of  $\{\beta_s\}_{s=1}^{1000}$  as a linearly decreasing sequence between  $\beta_1 = 0.2$  and  $\beta_{1000} = 10^{-4}$ .

**Measurement model** The measurement model was generated in the same way as for the GMM case.

**Posterior** The posterior samples were generated by running the No U-turn sampler ([Hoffman and Gelman, 2011]) with a chain of length  $10^4$  and taking the last sample of the chain. This was done in parallel to generate  $10^4$  samples. The mass matrix and learning rate were set by first running Stan's warmup and taking the last values of the warmup phase.

**Variational inference:** Variational inference in FMM shares the same details as the GMM case. The analogous of fig. 7 is displayed at fig. 13.

**Additional plots:** We now proceed to illustrate in Figures 14 to 16 the first 2 components for one of the measurement models for all the different combinations of  $(d_x, d_y)$  combinations used in table 1.

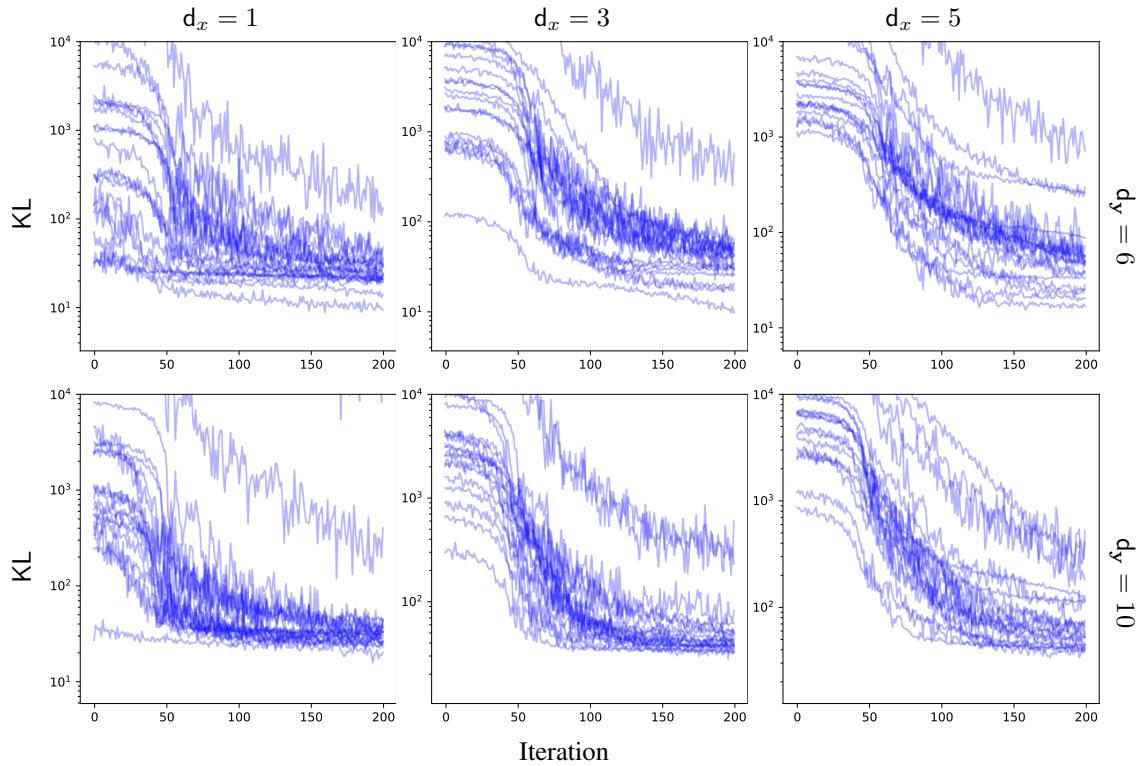


Figure 13: Evolution of KL with the number of iterations for all pairs of  $(d_x, d_y)$  tested in the FMM case.

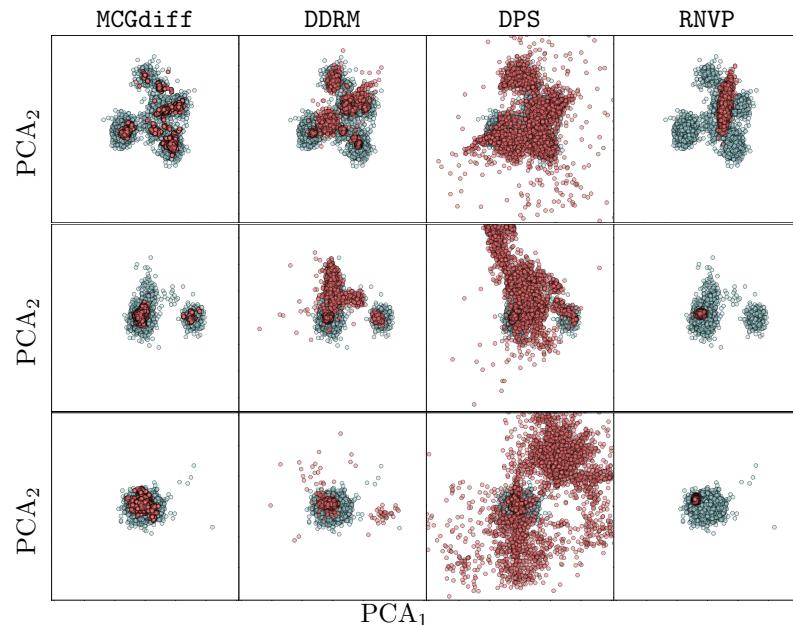


Figure 14: First two dimensions for the FMM case with  $d_x = 10$ . The rows represent  $d_y = 1, 3, 5$  respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).

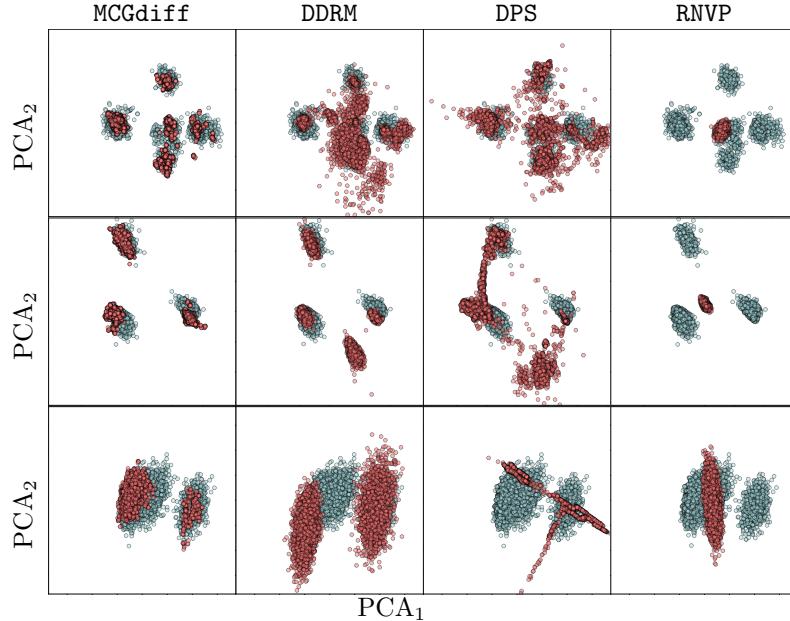


Figure 15: First two dimensions for the FMM case with  $d_x = 6$ . The rows represent  $d_y = 1, 3, 5$  respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).

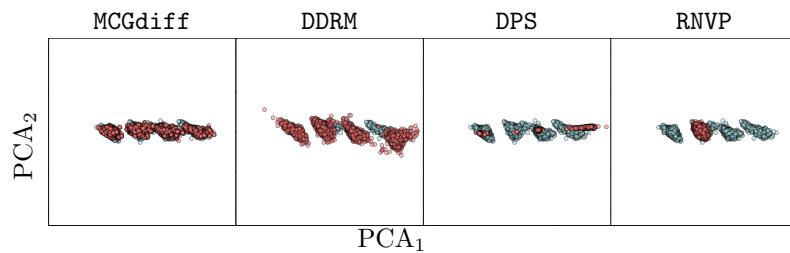


Figure 16: First two dimensions for the FMM case with  $d_x = 2$  and  $d_y = 1$ . The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).