

# Exact Evaluation of the Accuracy of Diffusion Models for Inverse Problems with Gaussian Data Distributions

Emile Pierret<sup>1\*</sup> and Bruno Galerne<sup>1,2</sup>

<sup>1</sup>Université d'Orléans, Université de Tours, CNRS, IDP, UMR 7013, Orléans, France.

<sup>2</sup>Institut universitaire de France (IUF).

\*Corresponding author(s). E-mail(s): [emile.pierret@univ-orleans.fr](mailto:emile.pierret@univ-orleans.fr);  
Contributing authors: [bruno.galerne@univ-orleans.fr](mailto:bruno.galerne@univ-orleans.fr);

## Abstract

Used as priors for Bayesian inverse problems, diffusion models have recently attracted considerable attention in the literature. Their flexibility and high variance enable them to generate multiple solutions for a given task, such as inpainting, super-resolution, and deblurring. However, several unresolved questions remain about how well they perform. In this article, we investigate the accuracy of these models when applied to a Gaussian data distribution for deblurring. Within this constrained context, we are able to precisely analyze the discrepancy between the theoretical resolution of inverse problems and their resolution obtained using diffusion models by computing the exact Wasserstein distance between the distribution of the diffusion model sampler and the ideal distribution of solutions to the inverse problem. Our findings allow for the comparison of different algorithms from the literature.

**Keywords:** Diffusion models, inverse problems, Gaussian distribution, conditional distribution, deblurring

## 1 Introduction

Inverse problems are ubiquitous in scientific imaging, where the goal is to reconstruct a clean image from partial or degraded observations. Such problems arise in a wide range of applications, including microscopy, medical imaging, computational photography, and satellite observation. Common tasks such as deblurring, super-resolution, and inpainting are typical examples. These problems are inherently ill-posed: multiple solutions are consistent with the observed data, making a single reconstruction often unreliable or unrepresentative of the underlying ambiguity.

A Bayesian framework offers a principled approach to handling this uncertainty. In this setting, observations are modeled as degraded

realizations from a prior distribution, and the objective becomes to characterize the posterior distribution of the clean image conditioned on these observations. This posterior encodes the full set of plausible solutions along with their associated uncertainties. The central challenge is thus to sample from this distribution in a faithful and efficient manner.

Generative models—particularly those trained on large datasets of natural images—have recently demonstrated remarkable capabilities in producing realistic samples. These include variational autoencoders (VAEs) [1, 2], generative adversarial networks (GANs) [3, 4], normalizing flows [5], and, more recently, diffusion models. Among these, diffusion models stand out for their training stability, their theoretically grounded formulation based on

stochastic processes, and their ability to generate perceptually high-quality samples [6]. In the context of inverse imaging problems, they have been successfully employed to produce visually convincing reconstructions that capture the diversity of admissible solutions [7–11], making these approaches the current state of the art.

However, despite their empirical success, a crucial question often remains overlooked: to what extent do the samples generated by these models faithfully reflect the true posterior distribution? This issue, already studied in the literature [12–14], is especially pressing in sensitive contexts, such as biomedical imaging or remote sensing, where biased or under-representative uncertainty estimates may have significant consequences. Common evaluation metrics, such as the Fréchet Inception Distance (FID) [15], are not suited for assessing statistical fidelity to the target posterior distribution. In this work, we directly compare image distributions.

In prior work [16], we studied diffusion models in their continuous formulation [17], focusing on Gaussian data distributions. While such a setting lacks direct practical relevance for real-world inverse problems, it provides a controlled and analytically tractable framework for evaluating the accuracy of diffusion-based posterior sampling. This Gaussian setting is also leveraged in recent theoretical studies to establish convergence and approximation guarantees for diffusion models [18, 19].

Building on these foundations, the present work focuses on the application of various diffusion-based algorithms from the literature to linear inverse problems involving images drawn from a Gaussian distribution. Under these assumptions, we are able to perform computations on low-dimensional toy examples and investigate the deblurring of Gaussian microtextures [20] at larger scales. Rather than relying on perceptual or empirical metrics, we propose a more rigorous analysis based on exact computation of Wasserstein distances directly between image distributions. This approach enables an exact quantitative assessment of the discrepancy between the generated distribution and the ground-truth posterior in a Gaussian framework where both quantities are explicitly accessible.

The remainder of the paper is organized as follows. In Section 2, we begin by reviewing the

discrete DDPM model [21], which serves as the basis for our analysis and then we introduce, within a unified framework, two posterior sampling algorithms from the literature: DPS [8] and IIGDM [10]. Next, in Section 3, under the assumption of Gaussian data, we present the Conditional Gaussian Diffusion Model (CGDM), an algorithm inspired by closed-form expressions available in this regime and we describe an efficient procedure for comparing these algorithms using the 2-Wasserstein distance, which we apply to several deblurring scenarios involving Gaussian microtextures in Section 4. We conclude with a discussion on the challenges of extending this methodology to broader classes of inverse problems in Section 5.

## 2 Reminder on diffusion models for solving inverse problems

### 2.1 Diffusion models for image generation

The goal of generative models is to sample a data distribution  $p_0$  of images. In this paper, we focus on the Discrete Denoising Diffusion Probabilistic Model (DDPM) [21] that consists in introducing first the forward process

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \mathbf{z}_t, \\ 1 \leq t \leq T, \quad \mathbf{z}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_0 \sim p_0, \end{aligned} \quad (1)$$

where  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  designates the standard normal distribution,  $T = 1000$  is the number of steps and  $(\beta_t)_{1 \leq t \leq T}$  is an increasing noise schedule. Ho *et al.* [21] propose a linear schedule from  $\beta_1 = 10^{-4}$  to  $\beta_T = 0.02$ , illustrated in Figure 1. All the transitions  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$  are Gaussian and by denoting  $p_t$  the density probability of  $\mathbf{x}_t$ ,  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ , for  $1 \leq t \leq T$ ,

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\xi}_t, \\ 1 \leq t \leq T, \quad \boldsymbol{\xi}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathbf{x}_0 \sim p_0. \end{aligned} \quad (2)$$

Consequently, by supposing that  $p_0$  admits an expectation  $\mu$  and a covariance matrix  $\Sigma$ ,

$$\mathbb{E}[\mathbf{x}_t] = \sqrt{\bar{\alpha}_t} \mu \quad (3)$$

$$\text{Cov}(\mathbf{x}_t) = \bar{\alpha}_t \Sigma + (1 - \bar{\alpha}_t) \mathbf{I}. \quad (4)$$

Note that  $\bar{\alpha}_t$  is decreasing such that  $\bar{\alpha}_T$  is close to 0 and the marginal distribution  $p_T$  of  $\mathbf{x}_T$  is close to  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . To define an approximate sampling procedure of the data distribution  $p_0$ , the objective is to reverse this process to go from  $\mathbf{x}_T$  to  $\mathbf{x}_0$ . The reverse process, called *backward process*, proposed by *Ho et al* [21] is the sequence of iterations

$$\begin{aligned}\mathbf{y}_T &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{y}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} (\mathbf{y}_t + \beta_t \nabla \log p_t(\mathbf{y}_t)) + \sigma_t \mathbf{z}_t, \quad (5) \\ \mathbf{z}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), 1 \leq t \leq T,\end{aligned}$$

where  $\nabla \log p_t$  is called the score function. Diffusion models are particularly used in the literature because the score function can be well estimated by a neural network (generally a U-Net model) by score matching [21, 22]. The two forward and backward processes are given in Algorithms 1 and 2.

**Remark 1** (Backward variance schedule). *The choice of  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$  with a diagonal covariance for the backward noise is optimal with  $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$  [22]. However, in [21], experimental results are similar with  $\sigma_t^2 = \beta_t$ . Another approach is to learn the noise schedule  $(\sigma_t)_{1 \leq t \leq T}$  in the form  $\exp(v \log \beta_t + (1-v) \log \tilde{\beta}_t)$  [6]. In the following, for simplicity we will take  $\sigma_t = \beta_t$  in our experiments but our results can easily be extended to the other variance schedules.*

## 2.2 DDPM for solving inverse problems

Let us recall some key aspects of diffusion models in the context of image restoration. We focus on solving linear inverse problems

$$\begin{aligned}\mathbf{v} &= \mathbf{A}\mathbf{x}_0 + \sigma\mathbf{n}, \\ \mathbf{x}_0 &\sim p_0, \sigma > 0, \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).\end{aligned}\quad (6)$$

In our context, we aim at sampling  $p_0(\cdot | \mathbf{v})$  to solve it. One can use a conditional DDPM associated with the following forward process

$$\begin{aligned}\tilde{\mathbf{x}}_t &= \sqrt{1-\beta_t} \tilde{\mathbf{x}}_{t-1} + \sqrt{\beta_t} \tilde{\mathbf{z}}_t, \\ 1 \leq t &\leq T, \quad \tilde{\mathbf{z}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{x}}_0 \sim p_0(\cdot | \mathbf{v}),\end{aligned}\quad (7)$$

and we denote by  $\tilde{p}_t$  the distribution of  $\tilde{\mathbf{x}}_t$  for  $0 \leq t \leq T$ . As before, the associated backward process is

$$\begin{aligned}\tilde{\mathbf{y}}_T &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \tilde{\mathbf{y}}_{t-1} &= \frac{1}{\sqrt{\alpha_t}} (\tilde{\mathbf{y}}_t + \beta_t \nabla \log \tilde{p}_t(\mathbf{y}_t)) + \beta_t \mathbf{z}_t, \\ \mathbf{z}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), 1 \leq t \leq T, \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\end{aligned}\quad (8)$$

Let us make the important following observation: Given  $\mathbf{x}_0$ ,  $\mathbf{x}_t$  is independent of  $\mathbf{v} = \mathbf{A}\mathbf{x}_0 + \sigma\mathbf{n}$ , so  $\tilde{p}_{t|0}(\tilde{\mathbf{x}}_t | \tilde{\mathbf{x}}_0) = p_{t|0}(\tilde{\mathbf{x}}_t | \tilde{\mathbf{x}}_0) = p_{t|0}(\tilde{\mathbf{x}}_t | \tilde{\mathbf{x}}_0, \mathbf{v})$  and

$$\tilde{p}_t(\tilde{\mathbf{x}}_t) = \int \tilde{p}_{t|0}(\tilde{\mathbf{x}}_t | \tilde{\mathbf{x}}_0) \tilde{p}_0(\tilde{\mathbf{x}}_0) d\tilde{\mathbf{x}}_0 \quad (9)$$

$$= \int p_{t|0}(\tilde{\mathbf{x}}_t | \mathbf{x}_0, \mathbf{v}) p_0(\mathbf{x}_0 | \mathbf{v}) d\mathbf{x}_0 \quad (10)$$

$$= p_t(\tilde{\mathbf{x}}_t | \mathbf{v}). \quad (11)$$

In other terms,

$$\tilde{p}_t = p_t(\cdot | \mathbf{v}), \quad 0 \leq t \leq T, \quad (12)$$

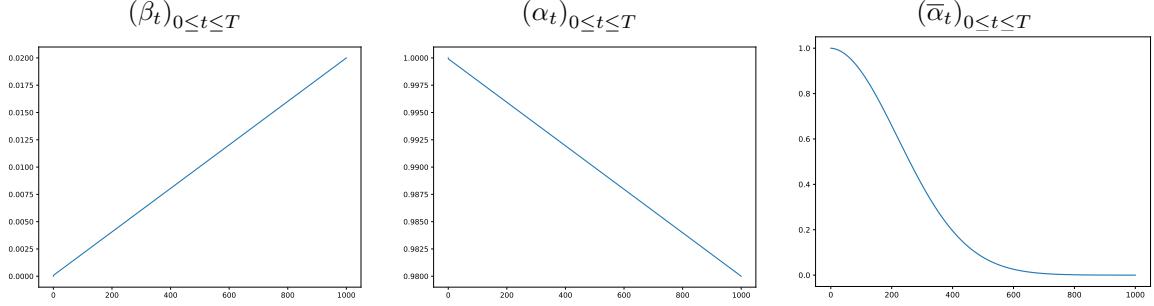
that is to say that it is equivalent to condition an unconditional forward process (1) on  $\mathbf{v}$  or consider a conditional forward process (7) to compute  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t | \mathbf{v})$ . Furthermore, by Bayes' rule,

$$\nabla_{\mathbf{x}} \log \tilde{p}_t(\mathbf{x}_t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}} \log p_t(\mathbf{v} | \mathbf{x}_t), \quad (13)$$

where  $p_t$  describes the unconditional forward process (Algorithm 1). In the following, we refer to  $p_t(\mathbf{v} | \mathbf{x}_t)$  as the **noisy likelihood** and to  $p_t(\mathbf{x}_t | \mathbf{v})$  as the **noisy posterior**.

Assuming that the score function  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  is well known and has already been applied in image generation, the goal is now to estimate the likelihood score  $\nabla_{\mathbf{x}} \log p_t(\mathbf{v} | \mathbf{x})$ . In practice, the noisy likelihood  $p_t(\mathbf{v} | \mathbf{x}_t)$  is generally intractable. To address this, several works [8, 10] assume that  $p_t(\mathbf{v} | \mathbf{x}_t)$  follows a Gaussian distribution, which can be fully characterized by its mean and covariance matrix. Importantly, the mean of  $p_t(\mathbf{v} | \mathbf{x}_t)$  is given by the following expression:

$$\mathbb{E}(\mathbf{v} | \mathbf{x}_t) = \mathbb{E}(\mathbf{A}\mathbf{x}_0 + \sigma^2 \mathbf{n} | \mathbf{x}_t) = \mathbf{A}\mathbb{E}(\mathbf{x}_0 | \mathbf{x}_t) \quad (14)$$



**Fig. 1: Illustration of the parameters for the DDPM model.** The sequence  $(\beta_t)_{0 \leq t \leq T}$  is taken linear from 0.0001 to 0.02, as done in [21]. In this case,  $T = 1000$  and  $\bar{\alpha}_T = 4.03E-5$ .

---

**Algorithm 1** DDPM forward process

---

```

1:  $x_0 \sim p_0$ 
2: for  $t=0$  to  $T-1$  do
3:    $\xi_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
4:    $x_{t+1} = \sqrt{1-\beta_t}x_t + \sqrt{\beta_t}\xi_t$ 
5: end for

```

---

where  $\mathbb{E}(x_0 | x_t)$  is the ideal MMSE denoiser. Moreover, by Tweedie's formula,

$$\begin{aligned} \hat{x}_0(x_t) &:= \mathbb{E}[x_0 | x_t] \\ &= \frac{1}{\sqrt{\alpha_t}}(x_t + (1 - \bar{\alpha}_t)\nabla_x \log p_t(x_t)) \end{aligned} \quad (15)$$

and the expectation  $\mathbb{E}[\mathbf{v} | x_t]$  is given by

$$\begin{aligned} \mathbb{E}[\mathbf{v} | x_t] &= A\hat{x}_0(x_t) \\ &= \frac{1}{\sqrt{\alpha_t}}A(x_t + (1 - \bar{\alpha}_t)\nabla_x \log p_t(x_t)). \end{aligned} \quad (16)$$

Then, it is necessary to choose a covariance matrix  $C_{\mathbf{v}|t}$  to approximate  $\text{Cov}(\mathbf{v} | x_t)$ . This results in

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{v} | x_t) = -\frac{1}{2}\nabla_{\mathbf{x}} \|\mathbf{v} - A\hat{x}_0(x_t)\|_{C_{\mathbf{v}|t}^{-1}}^2 \quad (17)$$

where we introduce the notation  $\|x\|_A = x^T Ax$  for a given positive symmetric matrix  $A$ . Given this model, an iteration of the conditional DDPM model becomes

$$\begin{aligned} \tilde{\mathbf{y}}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}(\tilde{\mathbf{y}}_t + \beta_t \nabla \log p_t(\tilde{\mathbf{y}}_t) - \frac{\beta_t}{2} \nabla_{\tilde{\mathbf{y}}_t} \|\mathbf{v} - A\hat{\mathbf{y}}_0(\tilde{\mathbf{y}}_t)\|_{C_{\mathbf{v}|t}^{-1}}^2) + \sigma_t \mathbf{z}_t, \\ \mathbf{z}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), 1 \leq t \leq T, \end{aligned} \quad (18)$$

as described in Algorithm 3.

---

**Algorithm 2** DDPM backward process

---

```

1:  $y_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t=T$  to 1 do
3:    $z_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
4:    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(y_t + \beta_t \nabla \log p_t(y_t)) + \sigma_t z_t$ 
5: end for

```

---



---

**Algorithm 3** Conditional backward DDPM process

---

**Require:**  $\mathbf{v}, (C_{\mathbf{v}|t})_{0 \leq t \leq T}$

```

1:  $y_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t=T$  to 1 do
3:    $\hat{\mathbf{y}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}}(y_t + (1 - \bar{\alpha}_t)\nabla \log p_t(y_t))$ 
4:    $\nabla \log p_t(\mathbf{y}_t | \mathbf{v}) = \nabla \log p_t(\mathbf{y}_t) - \frac{1}{2} \nabla_{\mathbf{y}_t} \|\mathbf{v} - A\hat{\mathbf{y}}_0(\mathbf{y}_t)\|_{C_{\mathbf{v}|t}^{-1}}^2$ 
5:    $z_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
6:    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{y}_t + \beta_t \nabla \log p_t(\mathbf{y}_t | \mathbf{v})) + \beta_t z_t$ 
7: end for

```

---

In the following, we concentrate on two algorithms proposed in the literature. Their respective parameterizations are detailed below and summarized in Table 1. Several other approaches can be found in the comprehensive survey by [7].

**The Diffusion Posterior Sampling (DPS).**

DPS is described in [8] to solve linear inverse problems such as inpainting, deblurring or super-resolution or nonlinear inverse problems such as phase retrieval or non-uniform deblurring. Chung *et al.* propose the following approximation

$$\log p_t(\mathbf{v} | x_t) \approx \log p(\mathbf{v} | x_0 = \hat{x}_0(x_t)) \quad (19)$$

As written in Equation (6),

$$p(\mathbf{v} | x_0) = \mathcal{N}(Ax_0, \sigma^2 \mathbf{I}). \quad (20)$$

Consequently, it is equivalent to fixing the covariance matrix of the noisy likelihood  $\mathbf{C}_{\mathbf{v}|t}^{\text{DPS}}$  to be equal to  $\sigma^2 \mathbf{I}$  and

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{v} | \mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)) \\ = -\frac{1}{2\sigma^2} \nabla_{\mathbf{x}_t} \|\mathbf{v} - A\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2.\end{aligned}\quad (21)$$

In practice, this method presents some instabilities. The choice of Chung *et al.* is equivalent to fixing

$$p(\mathbf{x}_0 | \mathbf{x}_t) \approx \delta_{\hat{\mathbf{x}}_0(\mathbf{x}_t)} \quad (22)$$

where  $\delta$  is a Dirac distribution and this could explain these instabilities: the variance of  $\mathbf{x}_0$  is neglected and consequently, applying the inverse of the underestimated covariance matrix  $\mathbf{C}_{\mathbf{v}|t}$  may cause the computations to diverge. To mitigate these instabilities, they introduce an hyperparameter  $\alpha_{\text{DPS}} > 0$  such that

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{v} | \mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)) \\ = -\frac{\alpha_{\text{DPS}}}{2\sigma^2} \nabla_{\mathbf{x}_t} \|\mathbf{v} - A\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2.\end{aligned}\quad (23)$$

$\alpha_{\text{DPS}}$  is both data and problem dependent (see [8], Appendix D.1). Finally, we consider  $\mathbf{C}_{\mathbf{v}|t}^{\text{DPS}} = \frac{\sigma^2}{\alpha_{\text{DPS}}} \mathbf{I}$ .

**Remark 2** (Gap between theory and practical implementation of the DPS algorithm). *In practice*, Chung et al. make a second approximation

$$\begin{aligned}\frac{\beta_t}{2\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log p(\mathbf{v} | \mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)) \\ = -\frac{\alpha_{\text{DPS}}}{\|A\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v}\|} \nabla_{\mathbf{x}_t} \|\mathbf{v} - A\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2 \\ = -\alpha_{\text{DPS}} \nabla_{\mathbf{x}_t} \|\mathbf{v} - A\hat{\mathbf{x}}_0(\mathbf{x}_t)\|\end{aligned}\quad (24)$$

This new formulation changes considerably the initial model and it amounts to put

$$\begin{aligned}\log p(\mathbf{v} | \mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t)) \\ \propto -\frac{2\sqrt{\alpha_t} \alpha_{\text{DPS}}}{\beta_t} \nabla_{\mathbf{x}_t} \|\mathbf{v} - A\hat{\mathbf{x}}_0(\mathbf{x}_t)\|.\end{aligned}\quad (25)$$

It can be interpreted as modeling the distribution  $p(\mathbf{v} | \mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_t))$  not as a Gaussian distribution but a modified Multivariate Generalized Gaussian Distribution (MGGD) [23, 24]. Another practical hint which is used in the official implementation of

this method<sup>1</sup> and that guarantees its stability is the clamping of the estimated denoised image  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$  between  $-1$  and  $1$ . To stay in the Gaussian realm, we do not consider these heuristic corrections in what follows.

### Pseudoinverse-Guided Diffusion Models (ΠGDM)

The ΠGDM algorithm [10] is described to solve inpainting, JPEG compression or deblurring problems. Song *et al.* make the following approximation

$$p(\mathbf{x}_0 | \mathbf{x}_t) \approx \mathcal{N}(\hat{\mathbf{x}}_0(\mathbf{x}_t), r_t^2 \mathbf{I}). \quad (26)$$

Consequently,

$$p(\mathbf{v} | \mathbf{x}_t) \approx \mathcal{N}(A\hat{\mathbf{x}}_0(\mathbf{x}_t), r_t^2 \mathbf{A}\mathbf{A}^T + \sigma^2 \mathbf{I}). \quad (27)$$

This is equivalent to choosing  $\mathbf{C}_{\mathbf{v}|t}^{\text{ΠGDM}} = r_t^2 \mathbf{A}\mathbf{A}^T + \sigma^2 \mathbf{I}$ , which now depends on the degradation operator  $\mathbf{A}$ . It is indeed natural for the degradation operator to appear, as well as a dependence on  $t$ . The hyperparameter  $r_t$  is estimated by considering the case where  $p_0$  is a standard normal distribution, which yields  $r_t^2 = 1 - \bar{\alpha}_t$  in the case of DDPM.

**Remark 3** (ΠGDM algorithm for DDPM). ΠGDM was first described for the DDIM algorithm [10]. However, the approximation of  $p_t(\mathbf{v} | \mathbf{x}_t)$  can be extended to the DDPM one.

Note that  $r_t^2 \mathbf{A}\mathbf{A}^T + \sigma^2 \mathbf{I}$  is invertible since  $\mathbf{A}\mathbf{A}^T$  is positive semi-definite. In the noiseless setting  $\sigma = 0$  (not considered here), a pseudo-inverse is applied, which is the reason this method is referred to as Pseudoinverse-Guided Diffusion Models.

## 3 Study under Gaussian assumption

The different algorithms using diffusion models are evaluated by computing empirical metrics on large datasets. The intractability of the score function and their conditional forms is a main obstacle to propose a theoretical study of their accuracy. In order to compare theoretically the algorithms, we will restrict to the case where

---

<sup>1</sup><https://github.com/DPS2022/diffusion-posterior-sampling>

$p_0$  is a Gaussian distribution by considering the following assumption.

**Assumption 1** (Gaussian assumption).  $p_0$  is a Gaussian distribution  $\mathcal{N}(\mu, \Sigma)$  in  $\mathbb{R}^d$ .

In this case, as developed below, we can derive all the closed-forms formulas of the distributions and precisely compare the different algorithms. Note that  $\Sigma$  is not assumed to be full rank, which includes the study of distributions defined on a manifold.

### 3.1 Exact Gaussian formulas

First, using a diffusion model to solve an inverse problem in the Gaussian case is generally unnecessary. In fact, we can explicitly derive the following conditional distribution

$$\begin{aligned} p(\mathbf{x}_0 | \mathbf{v}) &= \mathcal{N}\left(\boldsymbol{\mu}_{0|\mathbf{v}}, \mathbf{C}_{0|\mathbf{v}}\right) \\ \text{with } \boldsymbol{\mu}_{0|\mathbf{v}} &= \boldsymbol{\mu} + \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^{-1} (\mathbf{v} - \mathbf{A}\boldsymbol{\mu}), \quad (28) \\ \mathbf{C}_{0|\mathbf{v}} &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^{-1} \mathbf{A} \boldsymbol{\Sigma}, \\ \mathbf{M} &= \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T + \sigma^2 \mathbf{I}. \end{aligned}$$

and the proof is provided in Section A.3. As done in [25] for SR of Gaussian microtextures, we can then apply a kriging reasoning to sample  $\mathbf{x}_0$  conditionally to  $\mathbf{v}$ . In this context, the unconditional score  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$  is explicit and is given by

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = -\boldsymbol{\Sigma}_t^{-1} (\mathbf{x} - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}), \quad (29)$$

with

$$\boldsymbol{\Sigma}_t = \bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I}. \quad (30)$$

$\boldsymbol{\Sigma}_t$  is invertible for  $t > 0$ , as described for the continuous case in [16]. Given these closed expressions, we express exactly the conditional forward DDPM (Equation (7)) associated with  $\tilde{\mathbf{x}}_0 \sim p(\cdot | \mathbf{v})$  as

$$\begin{aligned} p_t(\tilde{\mathbf{x}}_t | \mathbf{v}) &= \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{t|\mathbf{v}}, \tilde{\mathbf{C}}_{t|\mathbf{v}}\right) \\ \text{with } \tilde{\boldsymbol{\mu}}_{t|\mathbf{v}} &= \sqrt{\bar{\alpha}_t} \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^\dagger (\mathbf{v} - \mathbf{A}\boldsymbol{\mu}), \\ \text{and } \tilde{\mathbf{C}}_{t|\mathbf{v}} &= \boldsymbol{\Sigma}_t - \bar{\alpha}_t \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^\dagger \mathbf{A} \boldsymbol{\Sigma}. \quad (31) \end{aligned}$$

In the Gaussian setting, the different algorithms have to be compared with this forward path which they are supposed to reconstruct along the time.

Another crucial derivation is the computation of the noisy likelihood  $p_t(\mathbf{v} | \mathbf{x}_t)$  that was modeled by a Gaussian distribution by DPS and IIGDM. In this particular case,  $p(\mathbf{v} | \mathbf{x}_t)$  is Gaussian without adding any assumption and can be expressed as

$$p_t(\mathbf{v} | \mathbf{x}_t) = \mathcal{N}\left(\mathbf{A} \hat{\mathbf{x}}_0(\mathbf{x}_t), (1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T + \sigma^2 \mathbf{I}\right), \quad (32)$$

$$\text{with } \hat{\mathbf{x}}_0(\mathbf{x}_t) = \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}). \quad (33)$$

The proofs are provided in Sections A.2 and A.4. Note that  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$  follows the Tweedie's formula (15). We consider the expression in the Gaussian case as corresponding to a new algorithm: Conditional Gaussian Diffusion Model (CGDM). We fix

$$\mathbf{C}_{t|\mathbf{v}}^{\text{CGDM}} = (1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T + \sigma^2 \mathbf{I} \quad (34)$$

which is the exact expression under the Gaussian assumption. Accordingly, the various algorithms are summarized in Table 1.

Let us observe the behavior of  $\mathbf{C}_{\mathbf{v}|t}$  for the different algorithms. Considering  $\alpha_{\text{DPS}} = 1$ , let us observe that for  $t$  close to  $T$ ,  $\boldsymbol{\Sigma}_t$  is close to  $\mathbf{I}$  and

$$\begin{aligned} \mathbf{C}_{\mathbf{v}|t}^{\text{DPS}} &= \sigma^2 \mathbf{I} \\ \mathbf{C}_{\mathbf{v}|t}^{\text{IIGDM}} &\approx (1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \sigma^2 \mathbf{I}, \\ \mathbf{C}_{\mathbf{v}|t}^{\text{CGDM}} &\approx (1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T + \sigma^2 \mathbf{I}. \end{aligned} \quad (35)$$

Consequently,  $\mathbf{C}_{\mathbf{v}|t}^{\text{IIGDM}}$  is closer to the exact theoretical expression  $\mathbf{C}_{\mathbf{v}|t}^{\text{CGDM}}$ , except that the prior covariance information is missing. The DPS algorithm significantly underestimates the covariance because  $1 - \bar{\alpha}_T$  is close to 1. For  $t$  close to 0,  $\boldsymbol{\Sigma}_t$  is close to  $\boldsymbol{\Sigma}$  and

$$\begin{aligned} \mathbf{C}_{\mathbf{v}|t}^{\text{DPS}} &= \sigma^2 \mathbf{I}, \\ \mathbf{C}_{\mathbf{v}|t}^{\text{IIGDM}} &\approx (1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \sigma^2 \mathbf{I}, \\ \mathbf{C}_{\mathbf{v}|t}^{\text{CGDM}} &\approx (1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T + \sigma^2 \mathbf{I}. \end{aligned} \quad (36)$$

Consequently,  $\mathbf{C}_{\mathbf{v}|t}^{\text{IIGDM}}$  is really close to the exact theoretical expression  $\mathbf{C}_{\mathbf{v}|t}^{\text{CGDM}}$  for low values of  $t$ . Let us note that the two expressions  $\mathbf{C}_{\mathbf{v}|t}^{\text{DPS}}, \mathbf{C}_{\mathbf{v}|t}^{\text{IIGDM}}$  are exact for  $t = 0$ . This is a key observation that will be important in practice.

	$\mathbf{C}_{\mathbf{v} t}$
DPS [8]	$\frac{\sigma^2}{\alpha_{\text{DPS}}} \mathbf{I}$
ΠGDM [10]	$(1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \sigma^2 \mathbf{I}$
CGDM	$(1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T + \sigma^2 \mathbf{I}, \quad \boldsymbol{\Sigma}_t = \bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I}$

**Table 1: Comparison of the exact expression of the likelihood score  $\nabla_{\mathbf{x}_t} \log p(\mathbf{v} | \mathbf{x}_t)$  (CGDM) with respect to the algorithmic models of DPS and ΠGDM.**  $\mathbf{C}_{\mathbf{v}|t}$  is such that the gradient  $\nabla_{\mathbf{x}_t} \log p(\mathbf{v} | \mathbf{x}_t)$  is modeled by  $-\frac{1}{2} \nabla_{\mathbf{x}_t} \| \mathbf{A} \hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v} \|_{\mathbf{C}_{\mathbf{v}|t}^{-1}}^2$ .

### 3.2 Comparison of the algorithms under Gaussian assumption

Here, we elucidate the structure of the covariance of the noisy posterior induced by each algorithm, given their respective choice of noisy likelihood. We then verify whether these correspond to a forward process.

#### Derivation of the noisy posterior $p_t(\mathbf{x}_t | \mathbf{v})$ for each algorithm

For each algorithm, we suppose that  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  is perfectly known and we use a model for  $\log p_t(\mathbf{v} | \mathbf{x}_t)$ . By using the Bayes' formula (Equation (13)) in a reverse sense than before, we can express a distribution  $p_t^{\text{algo}}(\mathbf{x}_t | \mathbf{v})$  verifying  $\log p_t^{\text{algo}}(\mathbf{x}_t | \mathbf{v}) = \log p_t(\mathbf{x}) + \log p_t^{\text{algo}}(\mathbf{v} | \mathbf{x}_t)$ . This noisy posterior is not related to the distributions sampled by the algorithms' backward processes unless it corresponds to a forward process (as discussed below), but it can still provide an interpretation of their model. By denoting  $p_t^{\text{algo}}(\mathbf{x}_t | \mathbf{v}) = \mathcal{N}(\boldsymbol{\mu}_{t|\mathbf{v}}^{\text{algo}}, \mathbf{C}_{t|\mathbf{v}}^{\text{algo}})$ , these computations lead to

$$\begin{aligned} \mathbf{C}_{t|\mathbf{v}}^{\text{DPS}} \\ = \mathbf{\Sigma}_t - \bar{\alpha}_t \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \bar{\alpha}_t \mathbf{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma} \end{aligned} \quad (37)$$

$$\begin{aligned} \mathbf{C}_{t|\mathbf{v}}^{\text{ΠGDM}} \\ = \mathbf{\Sigma}_t - \bar{\alpha}_t \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + (1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \bar{\alpha}_t \mathbf{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma} \end{aligned} \quad (38)$$

$$\begin{aligned} \mathbf{C}_{t|\mathbf{v}}^{\text{CGDM}} \\ = \mathbf{\Sigma}_t - \bar{\alpha}_t \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma}. \end{aligned} \quad (39)$$

All the details, with expressions of  $\boldsymbol{\mu}_{t|\mathbf{v}}^{\text{algo}}$  are given in Appendix A.6. We focus our discussions on the covariance matrices  $\mathbf{C}_{t|\mathbf{v}}^{\text{algo}}$  but similar observations can be established for the mean values

$\boldsymbol{\mu}_{t|\mathbf{v}}^{\text{algo}}$ . First, let us note that CGDM corresponds exactly to the forward distributions  $(\tilde{p}_t)_{0 \leq t \leq T}$  (see Equation (31)). Then, for  $t = 0$ , by supposing that  $\boldsymbol{\Sigma}$  is invertible,  $\bar{\alpha}_0 = 1$  and

$$\mathbf{C}_{0|\mathbf{v}}^{\text{DPS}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma} \quad (40)$$

$$\mathbf{C}_{0|\mathbf{v}}^{\text{ΠGDM}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + (1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \bar{\alpha}_t \mathbf{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma}. \quad (41)$$

These expressions are the exact covariance matrix of  $p(\mathbf{x}_0 | \mathbf{v})$ .

#### Algorithms studied in forward time evolution

Another interesting question is: Do  $p_t^{\text{algo}}(\mathbf{x}_t | \mathbf{v})$  corresponds to a forward DDPM process' distributions ? To correspond to a forward process,  $\mathbf{C}_{t|\mathbf{v}}$  is expected to be in the form

$$\mathbf{C}_{t|\mathbf{v}} = \bar{\alpha}_t \mathbf{C}_{0|\mathbf{v}} + (1 - \bar{\alpha}_t) \mathbf{I}. \quad (42)$$

As noted before,  $p_t^{\text{DPS}}(\mathbf{x}_t | \mathbf{v})$  has a covariance matrix

$$\begin{aligned} \mathbf{C}_{t|\mathbf{v}}^{\text{DPS}} \\ = \bar{\alpha}_t \left[ \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \bar{\alpha}_t \mathbf{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \boldsymbol{\Sigma} \right] + (1 - \bar{\alpha}_t) \mathbf{I} \end{aligned} \quad (43)$$

which does not correspond to a forward DDPM process in general because  $\sigma^2 \mathbf{I} + \bar{\alpha}_t \mathbf{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T$  depends on  $t$ . The only case in which this quantity does not depend on  $t$  is the trivial case where  $\mathbf{A} =$

0. For the IIGDM algorithm,

$$\begin{aligned} C_{t|v}^{\text{IIGDM}} &= \bar{\alpha}_t \left[ \Sigma - \Sigma A^T \left( \sigma^2 I + (1 - \bar{\alpha}_t) A A^T + \bar{\alpha}_t A \Sigma^2 \Sigma_t^{-1} A^T \right)^{-1} A \Sigma \right] \\ &\quad + (1 - \bar{\alpha}_t) I \end{aligned} \quad (44)$$

Similarly, this does not correspond to a standard forward DDPM process in general, since the expression  $\sigma^2 I + (1 - \bar{\alpha}_t) A A^T + \bar{\alpha}_t A \Sigma^2 \Sigma_t^{-1} A^T$  depends on  $t$ . Notably, in the case where  $\Sigma = I$ , we have  $C_{t|v}^{\text{IIGDM}} = C_{t|v}^{\text{CGDM}}$ . This is consistent with the fact that  $r_t^2$  was chosen in Section 2.2 to be exact in the case where  $p_0 = \mathcal{N}$ . For the CGDM algorithm, for any covariance matrix  $\Sigma$ ,

$$C_{t|v}^{\text{CGDM}} = \bar{\alpha}_t \left[ \Sigma - \Sigma A^T M^\dagger A \Sigma \right] + (1 - \bar{\alpha}_t) I \quad (45)$$

corresponds perfectly to the model forward (Equation (1)) applied to  $p_{0|v}$  (Equation (28)).

### 3.3 Recursive computation of the backward distributions

Each algorithm corresponds to a backward process, as given in Algorithm 3. We would like to characterize these at each time. In this Gaussian case, we can explicit  $-\frac{1}{2} \nabla_{\mathbf{x}_t} \|A\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v}\|_{C_{v|t}^{-1}}^2$ . In particular, the relation between  $\mathbf{x}_t$  and  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$  is linear, as given in Equation (33) and

$$\frac{1}{2} \nabla_{\mathbf{x}_t} \|A\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v}\|_{C_{v|t}^{-1}}^2 = \sqrt{\bar{\alpha}_t} \Sigma \Sigma_t^{-1} A^T C_{v|t}^{-1} (A\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v}). \quad (46)$$

As a consequence, the backward process of a given algorithm can be written as

$$\begin{aligned} \mathbf{y}_T &\sim \mathcal{N}(\mathbf{0}, I), \\ \mathbf{y}_{t-1} &= A_t^{\text{algo}} \mathbf{y}_t + b_t^{\text{algo}} + \beta_t \mathbf{z}_t, \quad 1 \leq t \leq T, \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, I) \end{aligned} \quad (47)$$

with

$$\begin{aligned} A_t^{\text{algo}} &= \frac{1}{\sqrt{\bar{\alpha}_t}} \left( I - \beta_t \Sigma_t^{-1} - \beta_t \bar{\alpha}_t \Sigma_t^{-1} \Sigma A^T (C_{v|t}^{\text{algo}})^{-1} A \Sigma \Sigma_t^{-1} \right), \end{aligned} \quad (48)$$

$$\begin{aligned} b_t^{\text{algo}} &= \beta_t \sqrt{\bar{\alpha}_{t-1}} \Sigma_t^{-1} \Sigma A^T (C_{v|t}^{\text{algo}})^{-1} (\mathbf{v} - A\mu + \bar{\alpha}_t A \Sigma \Sigma_t^{-1} \mu) \\ &\quad + \frac{\beta_t}{\sqrt{\bar{\alpha}_t}} \Sigma_t^{-1} \mu. \end{aligned} \quad (49)$$

This formulation implies that the corresponding backward processes remain Gaussian processes because all the operations are linear. To characterize it, it is necessary to compute

the means  $(\mu_t^{\text{algo}})_{0 \leq t \leq T}$  and covariance matrices  $(\Sigma_t^{\text{algo}})_{0 \leq t \leq T}$  at each time. In this Gaussian setting, since the score operations are linear, computing the means  $(\mu_t^{\text{algo}})_{0 \leq t \leq T}$  simply requires running the algorithms without adding noise at each step. The corresponding iterations are provided in Algorithm 4. To compute the covariance matrices  $(\Sigma_t^{\text{algo}})_{0 \leq t \leq T}$ , by using Equation (47),

$$\begin{aligned} \Sigma_T^{\text{algo}} &= I, \\ \Sigma_{t-1}^{\text{algo}} &= A_t^{\text{algo}} \Sigma_t^{\text{algo}} (A_t^{\text{algo}})^T + \beta_t I \end{aligned} \quad (50)$$

and it can be implemented by Algorithm 5. With these algorithms, we can characterize the algorithms' noisy posterior  $p_t^{\text{algo}}(\mathbf{x}_t | \mathbf{v})$  at each time and compare them to the forward process.

### 3.4 Comparison in terms of 2-Wasserstein distance

We established that under Gaussian assumption, the processes generated by DPS, IIGDM and CGDM are Gaussian with mean and covariance matrix iteratively computable by Algorithms 4 and 5. Consequently, we can compare these algorithms in terms of 2-Wasserstein distance which has a closed-form in this context [26]. For two Gaussian distributions  $\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)$ ,

$$\begin{aligned} \mathbf{W}_2^2(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) &= \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2} \Sigma_2 \Sigma_1^{1/2})^{1/2}). \end{aligned} \quad (51)$$

If in addition  $\Sigma_1$  and  $\Sigma_2$  are simultaneously diagonalizable with respective eigenvalues  $(\lambda_{i,1})_{1 \leq i \leq d}$  and  $(\lambda_{i,2})_{1 \leq i \leq d}$ ,

$$\begin{aligned} \mathbf{W}_2^2(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) &= \|\mu_1 - \mu_2\|^2 + \sum_{1 \leq i \leq d} \left( \sqrt{\lambda_{i,1}} - \sqrt{\lambda_{i,2}} \right)^2. \end{aligned} \quad (52)$$

#### Comparison of the noisy posteriors in toy models

We illustrate the comparison of the different algorithms in 2D and 3D in Figures 2 to 5. We study the inpainting problem which is conditioning on a noisy part of the coordinates of the Gaussian distribution. In order to highlight the differences between the algorithms, we consider in this section Gaussian distributions that are not scaled to lie

---

**Algorithm 4** Computation of the mean of the algorithm's backward along the time

---

```

1:  $\mu_T^{\text{algo}} \leftarrow \mathbf{0}$ 
2: for  $t=T$  to 1 do
3:    $\mu_{t-1}^{\text{algo}} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left( \mu_t^{\text{algo}} + \beta_t \nabla \log p_t(\mu_t^{\text{algo}} | \mathbf{v}) \right)$ 
4: end for

```

---

within the usual  $[-1, 1]$  range commonly used for images. In these examples, we compare the DPS, IIIGDM, and CGDM algorithms. Notably, CGDM aligns perfectly with the true theoretical distribution, even though the 2-Wasserstein distance is not zero. Indeed, we can note that the CGDM algorithm is not exact (by the observation of the 2-Wasserstein distance): it is affected by the incorrectness of the backward process. Theoretically, for a Gaussian distribution, the exact backward process is

$$\begin{aligned} \tilde{\mathbf{y}}_T &\sim \tilde{p}_T \\ \tilde{\mathbf{y}}_t &= \frac{1}{\sqrt{\alpha_t}} (\tilde{\mathbf{y}}_t + \beta_t \nabla \log \tilde{p}_t(\tilde{\mathbf{y}}_t)) + \sqrt{\beta_t} \mathbf{z}_t, \quad (53) \\ 1 \leq t &\leq T, \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_t^{-1} \Sigma_{t-1}). \end{aligned}$$

This formula is obtained in Appendix A.5. Consequently, two requirements are not fulfilled: First, the initialization is done with  $\tilde{\mathbf{y}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and not  $\tilde{\mathbf{y}}_T \sim \tilde{p}_T$ , which is known as the initialization error and discussed in [16]. Second, the added noise  $\mathbf{z}_t$  does not have the correct covariance matrix, it is not supposed to be diagonal. However, the 2-Wasserstein distance induced by these approximations is relatively low.

In Figure 2, the distributions along the time of the algorithms (2D bottom graphs) show that the DPS backward distribution moves into the space with false mean and covariance estimations along the time. The IIIGDM algorithm is very faithful to the theoretical backward in terms of mean but has not a perfect covariance information. These two facts are observable in the 2-Wasserstein distance graph: the 2-Wasserstein distance for CGDM remains consistently low, within the range of  $10^{-3}$  to  $10^{-2}$ , while IIIGDM varies between  $10^{-2}$  and  $10^{-1}$ . In contrast, DPS shows significantly higher deviation, reaching values above  $10^1$ , highlighting its instability and divergence from

---

**Algorithm 5** Computation of the covariance matrix of the algorithm's backward along the time

---

```

Require:  $(A_t^{\text{algo}})_{0 \leq t \leq T}$ 
1:  $\Sigma_T^{\text{algo}} = \mathbf{I}$ 
2: for  $t=T$  to 1 do
3:    $\Sigma_{t-1}^{\text{algo}} \leftarrow A_t^{\text{algo}} \Sigma_t^{\text{algo}} (A_t^{\text{algo}})^T + \beta_t \mathbf{I}$ 
4: end for

```

---

the true posterior distribution. Similar observations can be made in Figure 4. DPS is tested with  $\alpha_{\text{DPS}} = 0.2$  because it becomes unstable for higher values, although  $\alpha_{\text{DPS}} = 1$  would be the natural choice. At the final time  $t = 0$ , both DPS and IIIGDM exhibit a non-zero bias.

The ability of the DPS and IIIGDM algorithms to exhibit a decreasing error near the final time  $t = 0$  can be understood from the remarks in Section 3.2, as the expressions of the noisy posteriors for the different algorithms closely approximate the true noisy posterior when  $t$  is close to zero.

#### *Comparison of the estimated noisy likelihood $p_t(\mathbf{v} | \mathbf{x}_t)$*

Each algorithm is distinguished by its modeling choice for the covariance matrix of the noisy likelihood  $p_t(\mathbf{v} | \mathbf{x}_t)$  (see Table 1). In Figures 3 and 5, we compare these choices. More precisely, since  $p_t(\mathbf{v} | \mathbf{x}_t)$  depends on both  $\mathbf{x}_t$  and  $\mathbf{v}$ , where  $\mathbf{v}$  is related to  $\mathbf{x}_0$  via Equation (6), we proceed as follows: we first fix a sample  $\mathbf{x}_T$  drawn from the distribution  $p_T$  and compute the corresponding backward trajectory  $(\mathbf{x}_t)_{0 \leq t \leq T}$ . Then, we generate a noisy observation  $\mathbf{v}$  of the associated sample  $\mathbf{x}_0$ , and we plot  $p_t(\mathbf{v} | \mathbf{x}_t)$  at selected time steps. This allows us to visualize the model for  $\mathbf{C}_{\mathbf{v}|t}$  defined by each algorithm. In Figure 3, we observe that the variance of  $p_t(\mathbf{v} | \mathbf{x}_t)$  is significantly underestimated by the DPS algorithm. This may explain the instabilities observed in higher-dimensional settings: in this algorithm, the gradient term  $\nabla_{\mathbf{x}_t} \|\mathbf{A}\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2$  is divided by the low variance  $\sigma^2$ , amplifying its magnitude. Similarly, in Figure 5, the DPS algorithm again severely underestimates the variance and also introduces a substantial bias. The IIIGDM algorithm also suffers from inaccuracies in modeling  $p_t(\mathbf{v} | \mathbf{x}_t)$ , as its covariance model

$C_{v|t}^{\text{IIGDM}}$  does not incorporate the true covariance structure. However, the three algorithms are aligned at  $t = 0$ , as also discussed in Section 3.1.

## 4 Study case of the deblurring problem for Gaussian microtextures

### 4.1 The ADSN distribution and its covariance matrix

We consider the asymptotic discrete spot noise (ADSN) distribution [20] associated with an RGB texture  $\mathbf{u} \in \mathbb{R}^{3 \times \Omega_{M,N}}$  which is defined as the stationary Gaussian distribution whose covariance equals the autocorrelation of  $\mathbf{u}$ . More precisely, this distribution is sampled using convolution with a white Gaussian noise: Denoting  $m \in \mathbb{R}^3$  the channelwise mean of  $\mathbf{u}$  and  $\mathbf{t}_c = \frac{1}{\sqrt{MN}}(\mathbf{u}_c - m_c)$ ,  $1 \leq c \leq 3$ , its associated *texton*, for  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  of size  $M \times N$  the channelwise convolution

$$\mathbf{x} = m + \mathbf{t} \star \mathbf{w} \in \mathbb{R}^{3\Omega_{M,N}} \quad (54)$$

follows ADSN( $\mathbf{u}$ ) where  $\Omega_{M,N} = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ . This distribution is the Gaussian  $\mathcal{N}(m, \Sigma)$  where  $\Sigma$  is a RGB convolution ie in the form  $\Sigma = \mathbf{C}_t \mathbf{C}_t^T \in \mathbb{R}^{3\Omega_{M,N} \times 3\Omega_{M,N}}$  with  $\mathbf{C}_t := (\mathbf{C}_{t_1}^T \mathbf{C}_{t_2}^T \mathbf{C}_{t_3}^T)^T \in \mathbb{R}^{\Omega_{M,N} \times 3\Omega_{M,N}}$  where  $\mathbf{C}_{t_c} \in \mathbb{R}^{\Omega_{M,N} \times \Omega_{M,N}}$  is the convolution by the  $c$ -th channel of the texton  $\mathbf{t}$  for  $1 \leq c \leq 3$ . This correlation is induced by the fact that the white noise consider in Equation (54) is the same on each channel. In the Fourier domain, for  $\mathbf{x} \in \mathbb{R}^{3 \times \Omega_{M,N}}$ ,  $\xi \in \Omega_{M,N}$ ,

$$\widehat{\Sigma}x_i(\xi) = \widehat{\mathbf{t}}_i(\xi) \sum_{j=1}^3 \bar{\mathbf{t}}_j(\xi) \widehat{\mathbf{x}}(\xi) = \widehat{\mathbf{t}}(\xi) [\bar{\mathbf{t}}(\xi)]^T \widehat{\mathbf{x}}(\xi) \quad (55)$$

where  $\widehat{\mathbf{x}}$  is the Fourier transform of  $\mathbf{x}$  and  $\widehat{\mathbf{x}}(\xi) := (\widehat{x}_1(\xi) \ \widehat{x}_2(\xi) \ \widehat{x}_3(\xi))^T \in \mathbb{R}^3$ . Consequently, the matrix  $\Sigma$  acts as a rank-1 3D matrix on each frequency  $\xi \in \Omega_{M,N}$  of  $\mathbf{x}$  in the Fourier domain. Denoting by  $\widehat{\Sigma}(\xi)$  the action of the matrix  $\Sigma$  in the Fourier basis on the frequency  $\xi$ , we can write [27]

$$\widehat{\Sigma}(\xi) = \widehat{\mathbf{t}}(\xi) [\widehat{\mathbf{t}}(\xi)]^T. \quad (56)$$

We can provide the eigenvalues of  $\Sigma$ , that are  $\left(\sum_{i=1}^3 |\mathbf{t}_i(\xi)|^2\right)_{\xi \in \Omega_{M,N}}$  and 0 with multiplicity  $2MN$ . The score matrix  $\Sigma_t = \bar{\alpha}_t \Sigma + (1 - \bar{\alpha}_t) \mathbf{I}$  has the same structure as  $\Sigma$  and we can write

$$\widehat{\Sigma}_t(\xi) = \bar{\alpha}_t \widehat{\mathbf{t}}(\xi) [\widehat{\mathbf{t}}(\xi)]^T + (1 - \bar{\alpha}_t) \mathbf{I}_3. \quad (57)$$

As already done in [16], the score can be exactly applied in the context of Gaussian microtextures. The operations are detailed in Section B.1. As a consequence, we are able to implement a diffusion model with an exact score on ADSN microtextures, as illustrated in Figure 6. This direction is explored in the remainder of this section to analyze the DPS and IIGDM algorithms in the context of high-dimensional inverse problems. The covariance matrix structure will allow us to efficiently compute 2-Wasserstein distances in the context of deblurring.

### 4.2 Study of the deblurring problem

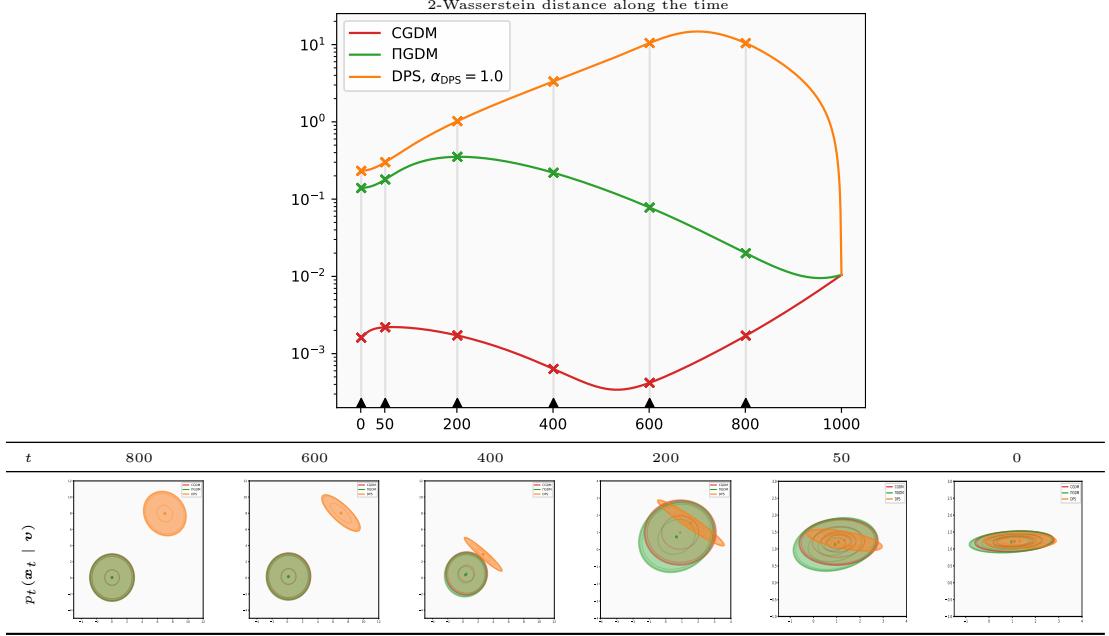
The degradation operator  $\mathbf{A}$  of the deblurring inverse problem is a channelwise convolution by a certain blur kernel  $\mathbf{c} \in \mathbb{R}^{\Omega_{M,N}}$ . In the following, we denote by  $\mathbf{C}$ , the block diagonal RGB convolution for which  $\mathbf{C}_c$  is on each block and where  $\mathbf{c} \in \mathbb{R}^{\Omega_{M,N}}$  is a blur kernel. In other terms,  $\mathbf{C}$  applies the same convolution by the blur kernel  $\mathbf{c}$  on each channel of an image. We focus on three blur kernels: the zoom-out bicubic kernel with a factor  $r = 16$ , which is the convolution part of the super-resolution (SR) problem and two motion blur kernels, generated by an online code<sup>2</sup>, which is also used in [8]. The effect of these degradations is illustrated in Figure 7.

In this specific problem for Gaussian microtextures, we have the following proposition, which allows us to compute efficiently the exact 2-Wasserstein distances associated with the different algorithms.

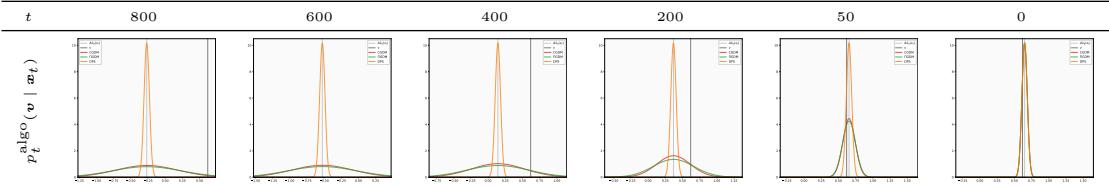
**Proposition 1** (Simultaneous diagonalizability of the Gaussian backward processes associated with the different algorithms). *For the deblurring problem involving ADSN microtextures, the covariance matrices associated with the backward processes of the different*

---

<sup>2</sup><https://github.com/LeviBorodenko/motionblur>



**Fig. 2: Illustration of the algorithms in 2D for the inpainting problem: Focus on the noisy posteriors  $p_t(\mathbf{x}_t \mid \mathbf{v})$ .** A 2D Gaussian distribution is conditioned on its first coordinate and noised at level  $\sigma = 10/255$ . Top: 2-Wasserstein distance along the time (from  $t = 1000$  to  $t = 0$ ) for the different algorithms with respect to the theoretical forward distribution. Bottom: For different times  $t$ , we plot the 2D noisy posterior of the algorithms at time  $t$ . Beware of the scale changes. Note the misalignment of the noisy posterior covariances at each time step.

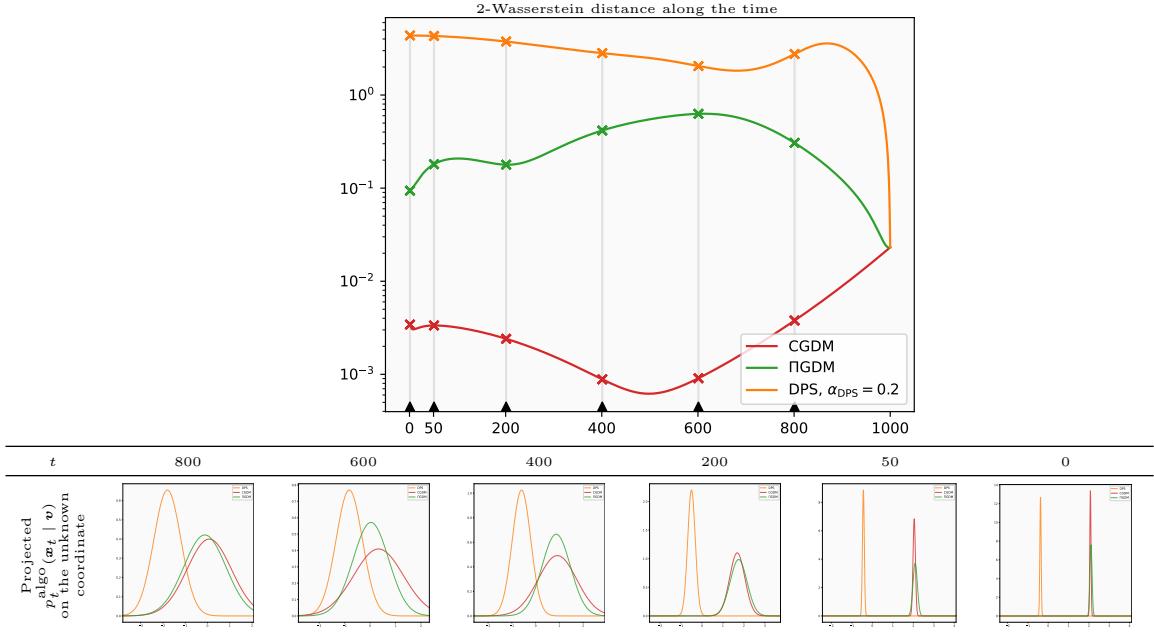


**Fig. 3: Illustration of the algorithms in 2D for the inpainting problem: Focus on the noisy likelihoods  $p_t(\mathbf{v} \mid \mathbf{x}_t)$ .** For different selected times  $t$ , we plot the 1D distribution model of  $p_t(\mathbf{v} \mid \mathbf{x}_t)$ , related to Figure 2. Beware of the scale changes. Note the underestimated variance in the DPS algorithm, and that all three algorithms coincide at  $t = 0$ .

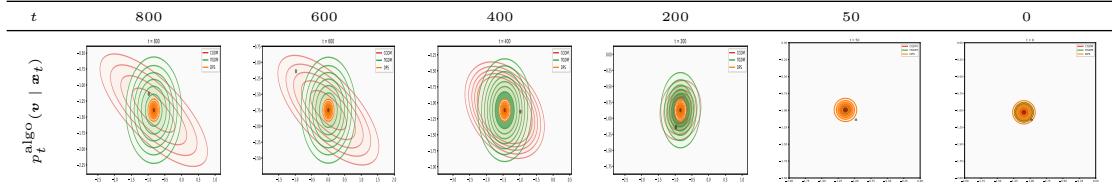
algorithms— $(\Sigma_t^{CGDM})_{0 \leq t \leq T}$ ,  $(\Sigma_t^{DPS})_{0 \leq t \leq T}$ , and  $(\Sigma_t^{\Pi GDM})_{0 \leq t \leq T}$ —are diagonalizable in the same orthogonal basis as  $\Sigma$ .

We provide a proof of this proposition in Section B.2. The main idea is that the degradation operator associated with the deblurring inverse problem preserves the structure of the covariance matrix of the ADSN models given in Equation (57). More precisely, to compute the

covariance component of the 2-Wasserstein distance (Equation 52), it suffices to consider only the eigenvalues of these covariance matrices at each time step, as done in [16] for the continuous case. This approach ensures fast and efficient computation of these distances.



**Fig. 4: Illustration of the algorithms in 3D for the inpainting problem: Focus on the noisy posteriors  $p_t(x_t | v)$ .** A 3D Gaussian distribution is conditioned on its two first coordinates and noised at level  $\sigma = 10/255$ . Top: 2-Wasserstein distance along the time for the different algorithms with respect to the theoretical forward distribution. Bottom: For different times  $t$ , we plot the 1D algorithms' backward distribution of the unknown coordinate at time  $t$ . Beware of the scale changes. We can observe the bias introduced by the DPS and ΠGDM algorithms over time.

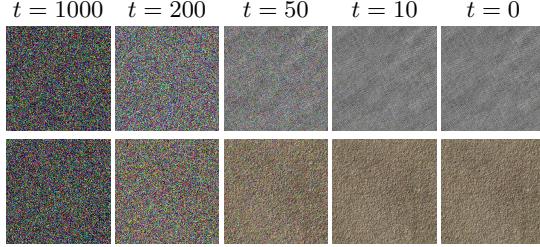


**Fig. 5: Illustration of the algorithms in 3D for the inpainting problem: Focus on the noisy likelihood  $p_t(v | x_t)$ .** A 3D Gaussian distribution is conditioned on its two first coordinates and noised at level  $\sigma = 10/255$ . For different times  $t$ , we plot the 2D distribution model of  $p_t(v | x_t)$ , related to Figure 4. Beware of the scale changes. The different algorithms exhibit alignment near the final time  $t = 0$ .

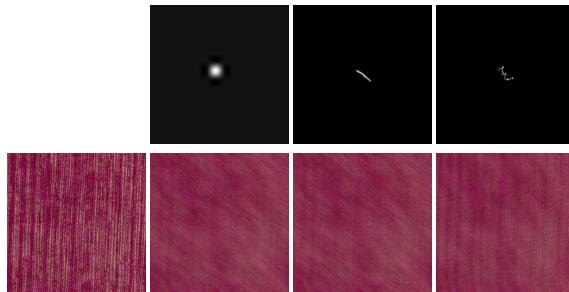
### 4.3 Numerical study of the deblurring problem: Exact computation of the 2-Wasserstein distance

We observe the exact 2-Wasserstein distances between the forward process and the backward processes generated by the different algorithms DPS, ΠGDM, CGDM for three different blur kernels examples with measurement noise level

$\sigma = 10/255$ , as illustrated in Figures 8 to 10. We use the exact unconditional score in this context (Equation (29)) and the different algorithm expression of Table 1. To emphasize the differences between the algorithms, we do not plot exactly the complete 2-Wasserstein distance between the distributions. We note that the 2-Wasserstein formula for simultaneously diagonalizable covariance matrices (Equation (52)) can be decomposed into two components. Denote by  $(\lambda_i, \Sigma)$   $1 \leq i \leq d$  the



**Fig. 6:** Illustration of the application of a DDPM on ADSN microtextures, with an exact score function.



**Fig. 7: Illustration of the effect of the studied blur kernels on images.** Top row: Bicubic kernel and two motion blur kernels. Bottom row: A clean image and its blurred versions corresponding to each kernel.

eigenvalues of  $\Sigma$ , where the null eigenvalues correspond to the kernel of  $\Sigma$ . We denote by  $\mathbf{P}_{\ker \Sigma}$  the orthogonal projection onto  $\ker \Sigma$ . For another matrix  $\Sigma_2$  that is diagonalizable in the same orthogonal basis, with eigenvalues  $(\lambda_{i,2})_{1 \leq d}$ ,

$$\mathbf{W}_2^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu_2, \Sigma_2)) \quad (58)$$

$$= \|\mu - \mu_2\|^2 + \sum_{1 \leq i \leq d} (\sqrt{\lambda_{i,\Sigma}} - \sqrt{\lambda_{i,2}})^2 \quad (59)$$

$$= \underbrace{\|\mathbf{P}_{\ker \Sigma}(\mu - \mu_2)\|^2 + \sum_{\lambda_{\Sigma,i}=0} (\sqrt{\lambda_{i,2}})^2}_{:= \mathbf{W}_{2,\ker \Sigma}^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu_2, \Sigma_2))} \quad (60)$$

$$+ \underbrace{\|(\mathbf{I} - \mathbf{P}_{\ker \Sigma})(\mu - \mu_2)\|^2 + \sum_{\lambda_{\Sigma,i} \neq 0} (\sqrt{\lambda_{i,\Sigma}} - \sqrt{\lambda_{i,2}})^2}_{:= \mathbf{W}_{2,(\ker \Sigma)^\perp}^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu_2, \Sigma_2))}. \quad (61)$$

In Figures 8 to 10, we plot  $\mathbf{W}_{2,(\ker \Sigma)^\perp}$ . Let us discuss why this metric is of particular interest by

showing that  $\mathbf{W}_{2,\ker \Sigma}$  is identical across all algorithms. Indeed, on  $\ker \Sigma$ , we can precisely describe the iterations of the different algorithms. All algorithms behave identically on the zero eigenvalues of  $\Sigma$ , acting like an unconditional DDPM within this kernel subspace. This can be seen by considering the operators  $\mathbf{A}_t$  and vectors  $\mathbf{b}_t$  defined in Section 3.3. For  $1 \leq t \leq T$ ,

$$\mathbf{P}_{\ker \Sigma} \mathbf{A}_t = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{P}_{\ker \Sigma} - \beta_t \mathbf{P}_{\ker \Sigma} \Sigma_t^{-1} \right) \quad (62)$$

$$= \frac{1}{\sqrt{\alpha_t}} \mathbf{P}_{\ker \Sigma} \left( \mathbf{I} - \beta_t \frac{1}{1 - \bar{\alpha}_t} \mathbf{I} \right), \quad (63)$$

$$\mathbf{P}_{\ker \Sigma} \mathbf{b}_t + \frac{\beta_t}{\sqrt{\alpha_t}} \mathbf{P}_{\ker \Sigma} \Sigma_t^{-1} \mu \quad (64)$$

$$= \mathbf{P}_{\ker \Sigma} \frac{\beta_t}{\sqrt{\alpha_t}} \frac{1}{1 - \bar{\alpha}_t} \mu. \quad (65)$$

These expressions are independent of  $\mathbf{C}_{v|t}$ , the covariance matrix that distinguishes the different algorithms. As a consequence, the 2-Wasserstein on  $\ker \Sigma$  error is

$$\begin{aligned} & \mathbf{W}_{2,\ker \Sigma}^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu_2, \Sigma_2)) \\ & := \|\mathbf{P}_{\ker \Sigma}(\mu - \mu_2)\|^2 + (\dim \ker) \times \mathcal{E}_0(t) \end{aligned} \quad (66)$$

where  $\mathcal{E}_0(t)$  is the error provided by the DDPM algorithm, with an exact score on zero eigenvalues at time  $t$ . This error is due to the fact that the DDPM scheme is unable to retrieve the rank of  $\Sigma$  (that has also been observed for the discrete schemes in [16]). Note that this error can be managed by modifying the noise schedule, as explored in [18].

Let now discuss the metric of interest  $\mathbf{W}_{2,(\ker \Sigma)^\perp}$ . The hyperparameter  $\alpha_{DPS}$  respects the following rule: the lowest stable value provides the lowest 2-Wasserstein distance and we use it in Figures 8 to 10. The algorithms are ranked in this order in terms of performance: CGDM, ΠIGDM and DPS along the time. The superior performance of CGDM was expected in this Gaussian setting. For the bicubic kernel and each texture, the ΠIGDM is quite close to the perfect CGDM algorithm at the end of the process while DPS presents a high 2-Wasserstein error along all the iterations. For the two motion blur kernels, ΠIGDM stays relatively far from the true conditional algorithm.

Samples shown in Figures 11 and 13 for two texture examples show that the samples generated by the different algorithms seem very similar. However, the corresponding mean demonstrates that the two algorithms are significantly biased,

especially the DPS one. Figure 12 show the mean along the backward process in for one texture example and as observed in our toy examples in 2D and 3D, the DPS and IIGDM algorithms are biased, and particularly DPS.

We compare the models of the noisy likelihood covariance of  $p_t(\mathbf{v} \mid \mathbf{x}_t)$  in Figure 14 for the first motion kernel, for the first fabric texture. We can observe that the constant DPS is really far from the true theoretical distribution while IIGDM approximation becomes less harmful along the time, it can be explained by our empirical observations in Section 3.2.

However, the covariance conditional distribution are very close for the three algorithms for lower times, as illustrated in Figure 15, which is related to our observations in Section 3.2: For  $t$  close to 0, the algorithms converge towards the correct conditional distribution  $p_0(\cdot \mid \mathbf{v})$ .

## 5 Discussion

The bias induced by the two algorithms, DPS and IIGDM, raises questions about their suitability for uncertainty quantification. However, despite these advantages, it is important to note that CGDM is significantly more computationally expensive than IIGDM. The exact Gaussian computations required by CGDM introduce a higher complexity, which may prevent its deployment in large-scale. On the other hand, IIGDM, while slightly less accurate, provides a much more computationally efficient alternative, making it the preferred method in practical scenarios where the covariance matrix cannot be quickly invertible (see the case of super-resolution below). As a result, IIGDM appears to be the go-to approach for most real-world applications of conditional diffusion models, striking a balance between accuracy and computational feasibility. We discuss below the extension of our study of the CGDM algorithms to more general inverse problems.

### *Extension to the SR inverse problem for the Gaussian microtextures*

Let us discuss the extension of our work to non diagonal inverse problems with a focus on the super-resolution (SR) problem. Let us consider  $\mathbf{A} = \mathbf{SC}$  where  $\mathbf{S}$  is a subsampling operator with stride  $r \in \mathbb{N}$  and  $\mathbf{C}$  is a convolution operator. The conditional sampling of this inverse problem has

been considered and solved in [25] by a kriging reasoning for Gaussian microtextures. We illustrate preliminary results we obtain in this case in Figure 16. We can compute the backward mean evolution of the different algorithms by applying their backward steps without adding noise, as illustrated in Algorithm 4. We can observe the bias evolution of the different algorithms and we observe similar results than in the deblurring case, ranking methods in this order: CGDM, IIGDM and DPS. We observe that DPS is much more stable, it is possible to take  $\alpha_{\text{DPS}} = 1$ . However, the extension of the whole previous reasoning on deblurring to this problem is not trivial for the reasons explained below.

### *Inability to compute the likelihood $\nabla \log p(\mathbf{v} \mid \mathbf{x}_t)$ term for RGB images.*

For CGDM, we need to compute  $(\sigma^2 \mathbf{I} + \mathbf{SC}\Sigma_t^{-1}\mathbf{C}^T\mathbf{S}^T)^{-1}$ . As demonstrated in [16], computing efficiently and stably this inverse is a hard issue but it can be well-approximated by a diagonal RGB convolution, as done in Figure 16 and explained in [16].

### *Non-simultaneous diagonalizability of the different algorithms along the time.*

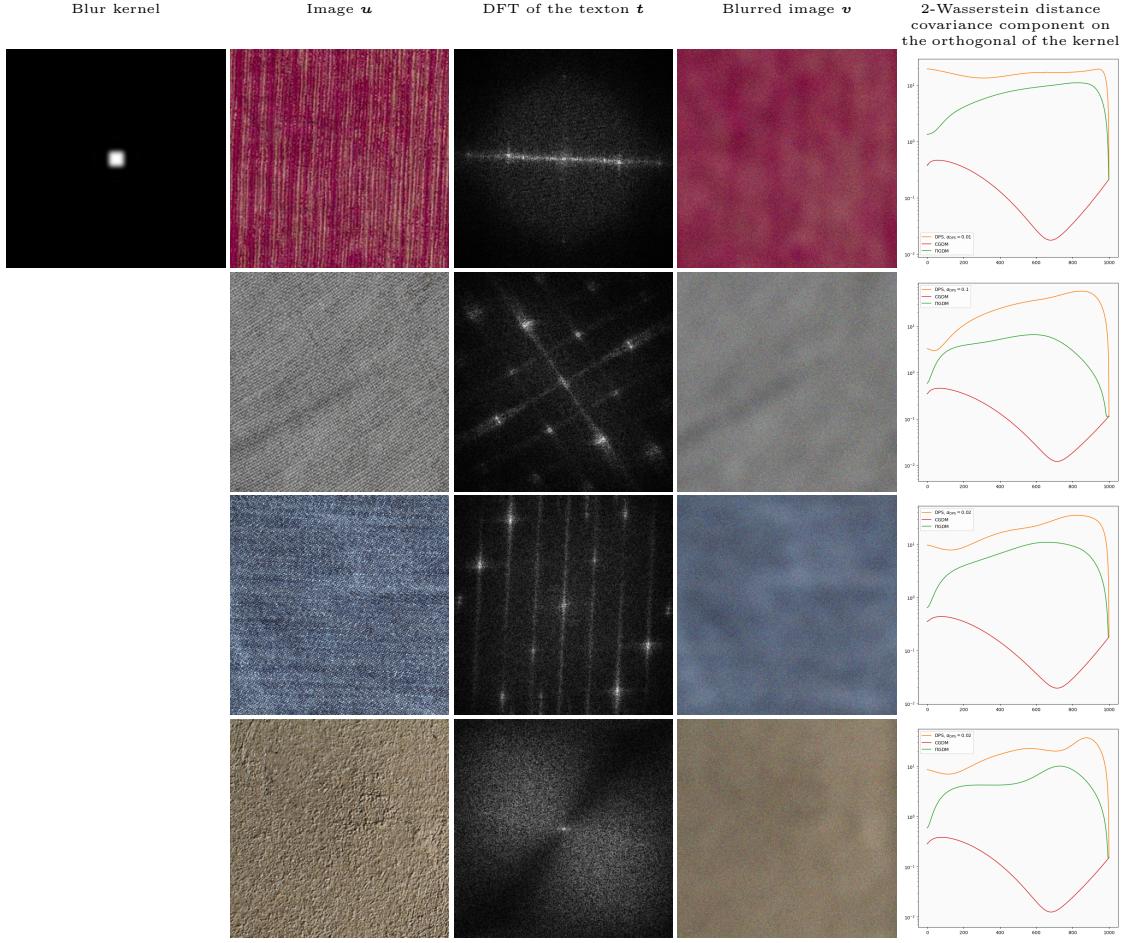
All the covariance matrices  $(\Sigma_t^{\text{DPS}})_{0 \leq t \leq T}$ ,  $(\Sigma_t^{\text{IIGDM}})_{0 \leq t \leq T}$ ,  $(\Sigma_t^{\text{CGDM}})_{0 \leq t \leq T}$  are not simultaneously diagonalizable. Indeed, let observe the operator  $\mathbf{A}_T^{\text{DPS}}$  defined in Section 3.3, with  $\alpha_{\text{DPS}} = 1$ .

$$\mathbf{A}_T = \frac{1}{\sqrt{\alpha_T}} \left( I - \beta_T \Sigma_t^{-1} - \frac{\beta_T}{\sigma^2} \bar{\alpha}_T \Sigma_T^{-1} \Sigma \mathbf{C}^T \mathbf{S}^T \mathbf{S} \mathbf{C} \Sigma \Sigma_T^{-1} \right) \quad (67)$$

In particular, the operator  $\mathbf{S}^T \mathbf{S}$  is not diagonalizable in the Fourier domain and breaks the Fourier structure of the covariance matrices. As discussed in Section B.2, this also applies to blur kernels that are not identical across channels.

### *Instabilities in high dimension.*

The previous issue could be overcome by using the general expression of the 2-Wasserstein metric (Equation (51)), which does not rely on simultaneous diagonalization. Nonetheless, the positive symmetric square root matrix of  $\Sigma$  size has to be computed. The size of this matrix is  $(3MN)^2$  which is too high to be computed in practice.



**Fig. 8: 2-Wasserstein distance evolution of the different algorithms for the bicubic kernel.** From left to right: log modulus of the DFT of the blur kernel, image  $\mathbf{u}$  associated with the ADSN distribution, log modulus of the DFT of the texton  $\mathbf{t}$ , blurred image  $\mathbf{v}$ , 2-Wasserstein distance of the different algorithms with respect to the forward process, along the time. We observe a consistent ranking of the algorithms in terms of performance—DPS, IIGDM, and CGDM—from lowest to highest, across all kernels and throughout the diffusion process.

Instabilities increase with the applications of it during 1000 steps.

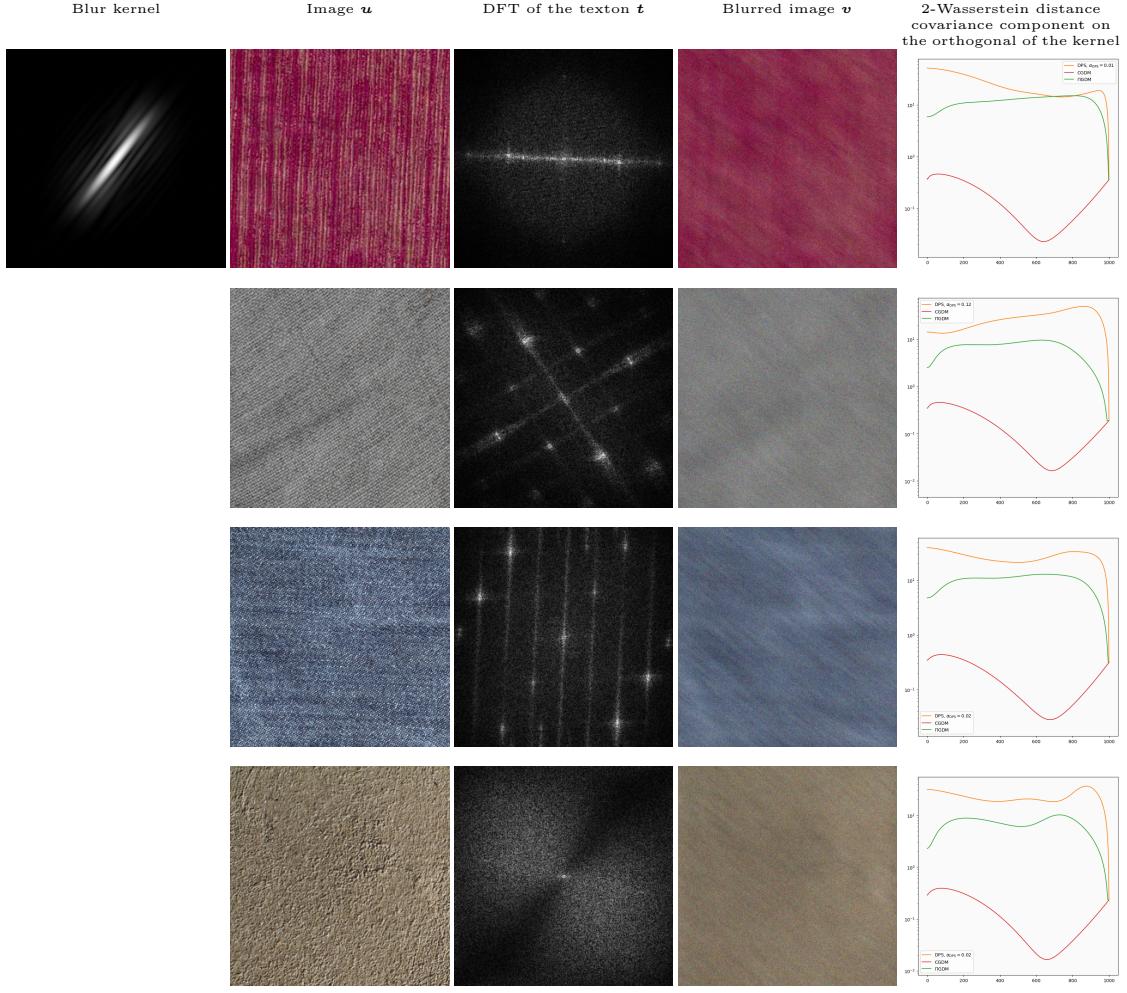
#### Discussion on other inverse problems

The inability to compute  $\nabla \log p(\mathbf{v} \mid \mathbf{x}_t)$  along time, combined with the lack of simultaneous diagonalizability across different algorithms, also applies to more general inverse problems such as inpainting. In general, there is no reason to expect that the data covariance and the degradation operator share the same eigenvectors [28]. This presents a challenging research direction for

developing metrics that closely approximate the exact 2-Wasserstein distance.

## 6 Conclusion

We presented a rigorous evaluation of conditional diffusion models under Gaussian priors for inverse problems, with exact 2-Wasserstein computations in deblurring tasks. Our results show that both DPS and IIGDM exhibit notable biases and fail to adequately capture the posterior distribution, while our proposed CGDM aligns more closely with the true conditional law. Although IIGDM

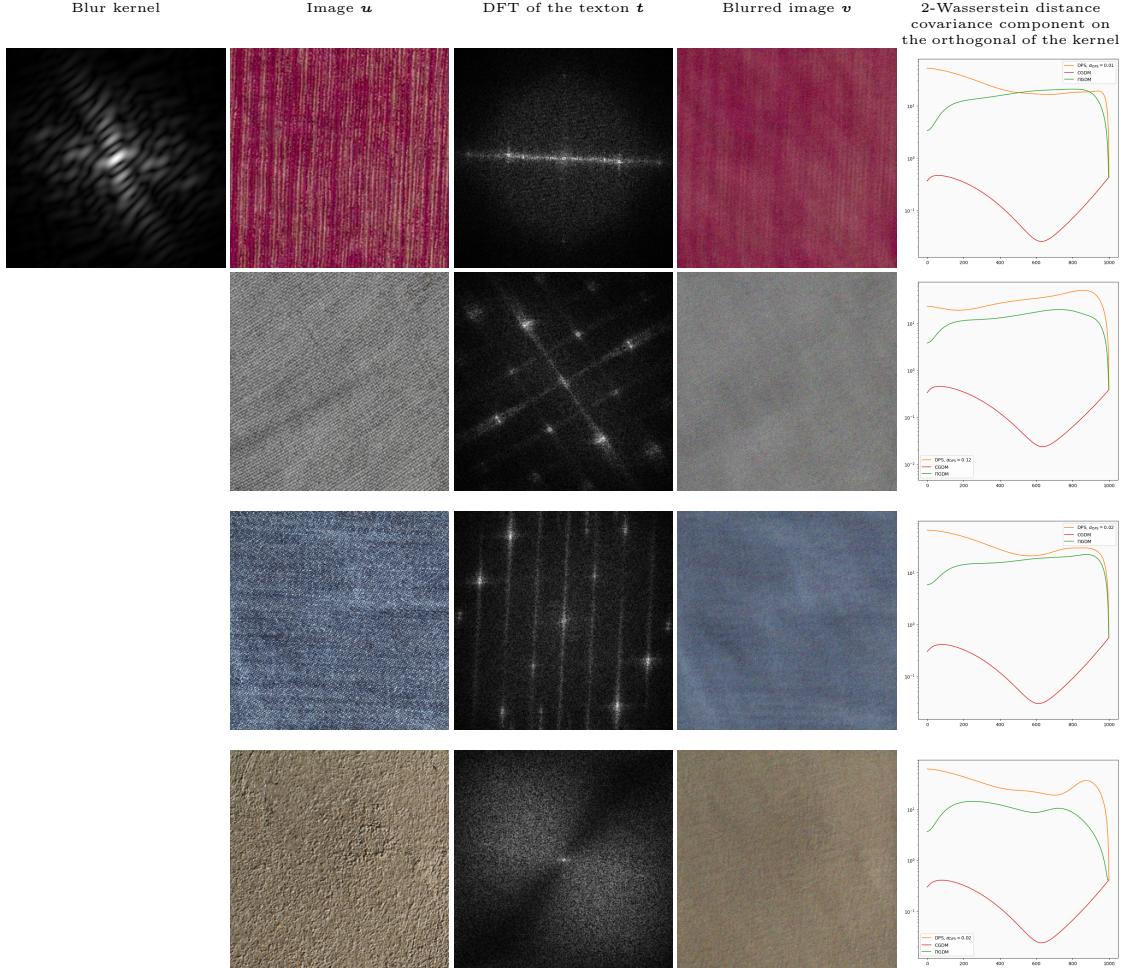


**Fig. 9: 2-Wasserstein distance evolution of the different algorithms for the first motion blur kernel.** From left to right: log modulus of the DFT of the blur kernel, image  $\mathbf{u}$  associated with the ADSN distribution, log modulus of the DFT of the texton  $\mathbf{t}$ , blurred image  $\mathbf{v}$ , 2-Wasserstein distance of the different algorithms with respect to the forward process, along the time. We observe a consistent ranking of the algorithms in terms of performance—DPS,  $\Pi$ GDM, and CGDM—from lowest to highest, across all kernels and throughout the diffusion process, except for the first example around intermediate time steps.

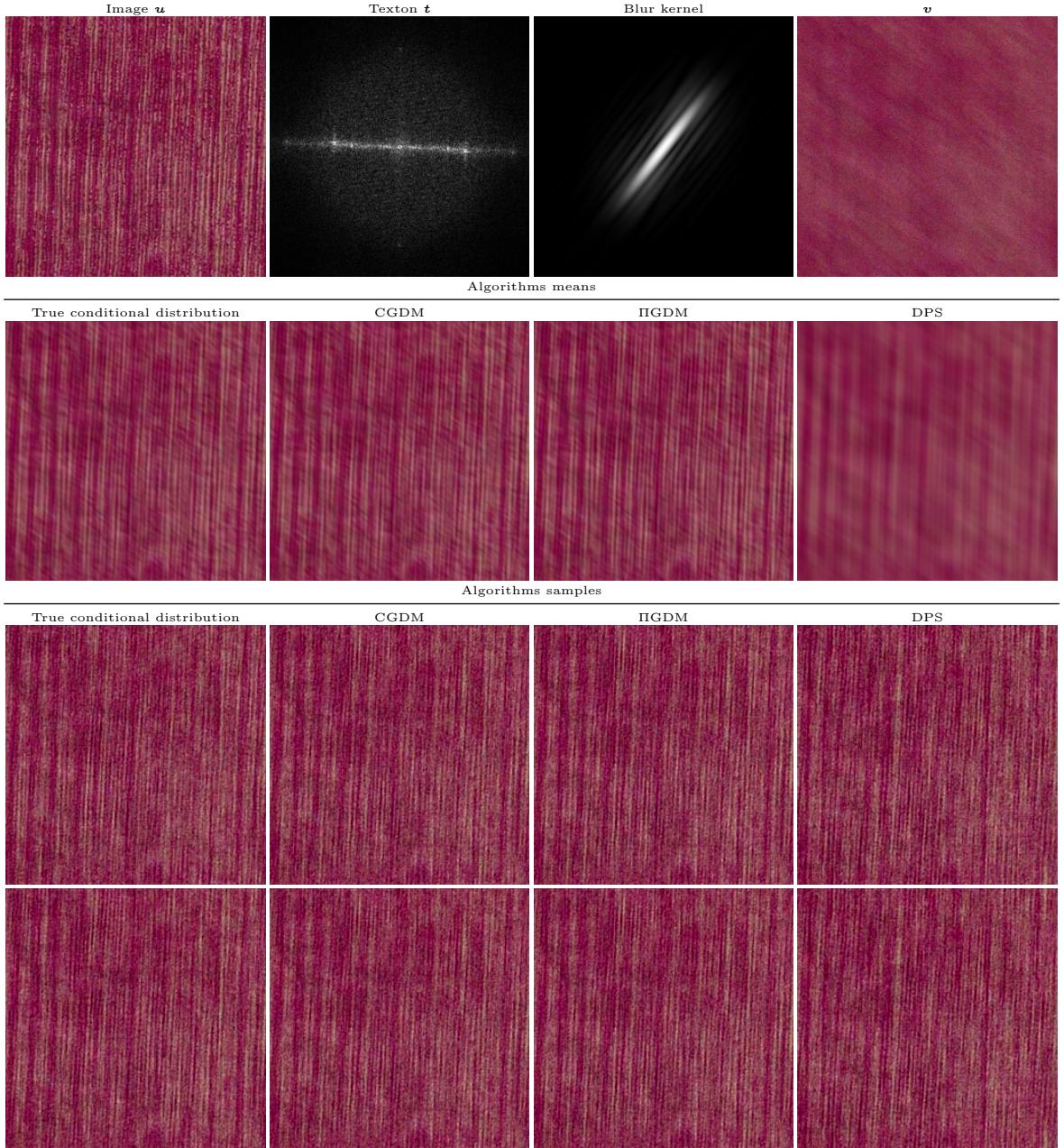
is computationally faster, CGDM achieves a more faithful approximation of the posterior. Beyond deblurring, our methodology could be extended to a broader range of inverse problems and more complex, non-Gaussian data distributions. Extending this framework to such settings raises important open questions for future research.

## Acknowledgements

The authors acknowledge the support of the project MISTIC (ANR-19-CE40-005).



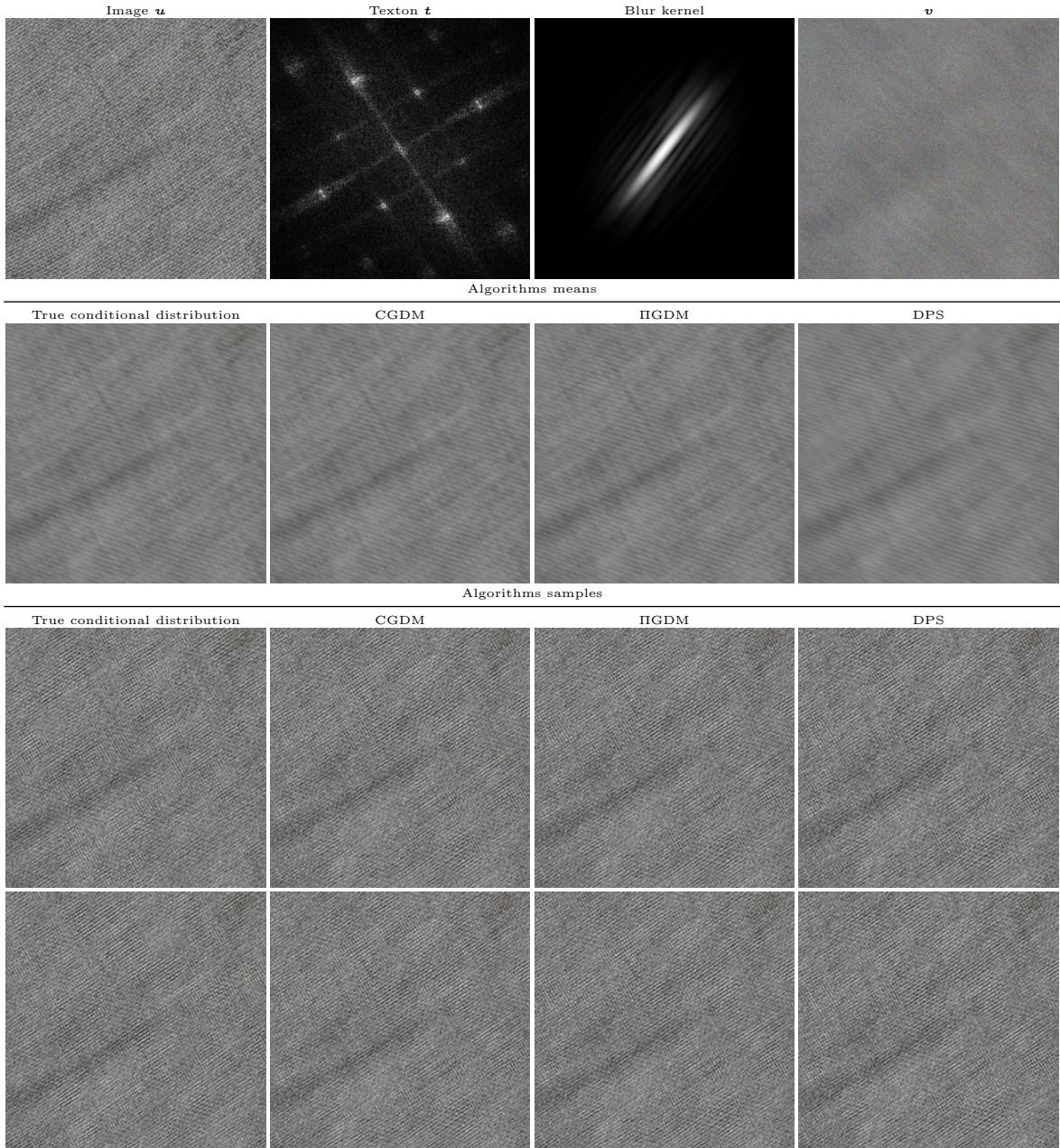
**Fig. 10: 2-Wasserstein distance evolution of the different algorithms for the second motion blur kernel.** From left to right: log modulus of the DFT of the blur kernel, image  $\mathbf{u}$  associated with the ADSN distribution, log modulus of the DFT of the texton  $\mathbf{t}$ , blurred image  $\mathbf{v}$ , 2-Wasserstein distance of the different algorithms with respect to the forward process, along the time. We observe a consistent ranking of the algorithms in terms of performance—DPS,  $\Pi$ GDM, and CGDM—from lowest to highest, across all kernels and throughout the diffusion process, except for the first example around intermediate time steps.



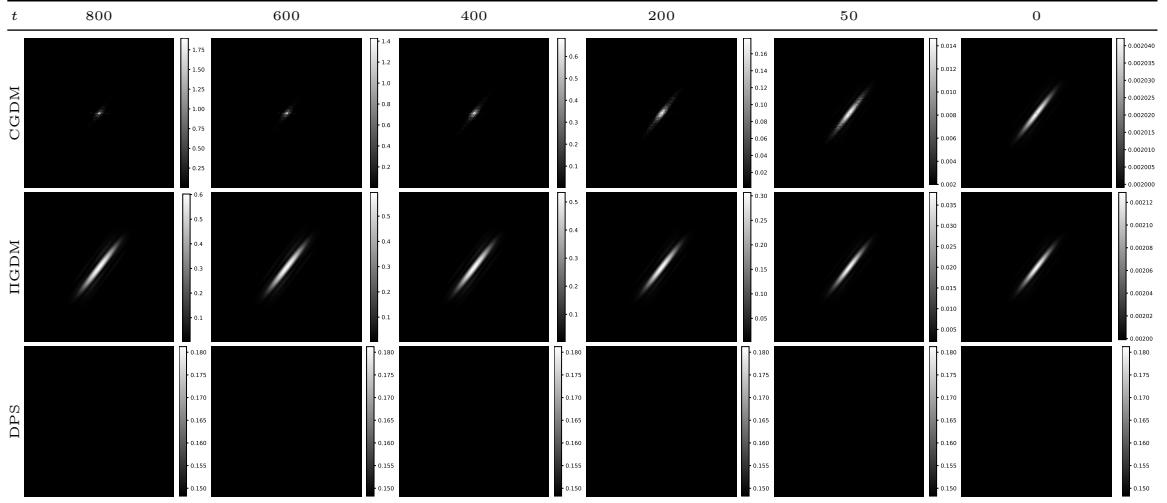
**Fig. 11: Samples and means generated by the different algorithms.** CGDM, IIIGDM, DPS samples are generated from the same seed at each step and are very similar. However, the mean of the DPS algorithm contains less texture information than the other algorithms which illustrates its bias.



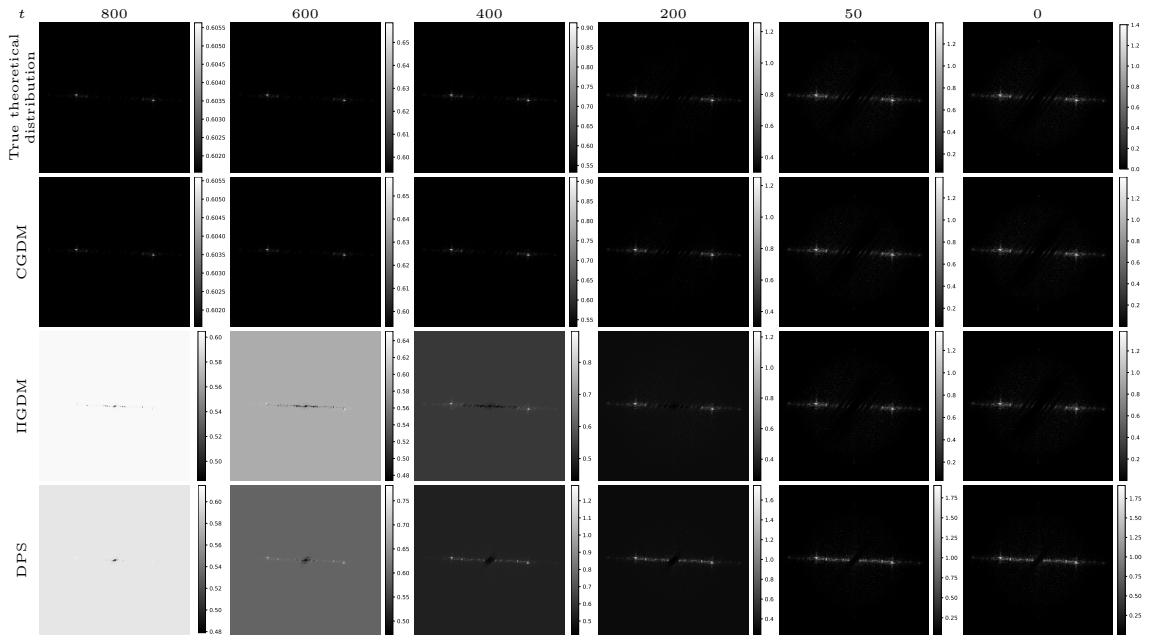
**Fig. 12: Means of the algorithms along the time.** It corresponds to the first motion blur kernel, for the first fabric texture in Figure 8. Note that the DPS algorithm suffers from a relative important bias along times, as observed for Gaussian distributions in small dimension.



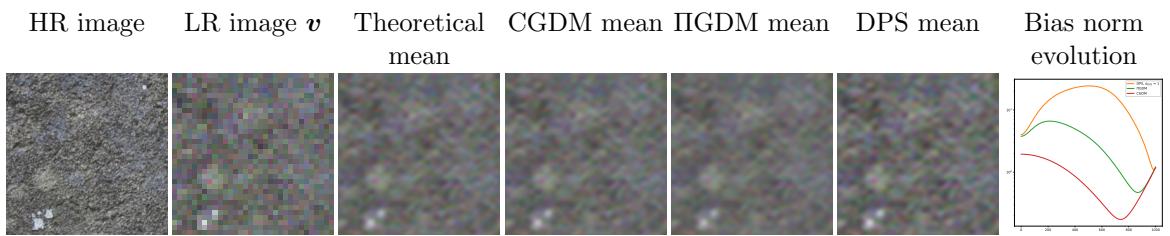
**Fig. 13: Samples and means generated by the different algorithms.** CGDM, IIIGDM, DPS samples are generated from the same noise at each step and are very similar. However, the means are perceptually really different.



**Fig. 14:** Model chosen by the different algorithm for the noisy likelihood  $p_t(\mathbf{v} | \mathbf{x}_t)$ . DFT of the kernel associated with the distribution  $p_t(\mathbf{v} | \mathbf{x}_t)$  for the different algorithms at different times for the bicubic kernel, for the first fabric texture in Figure 8. Note that IIIGDM incorporates the initial motion blur information, whereas the DPS kernel remains constant. CGDM also accounts for the texton information, although the kernel is not perfectly represented.



**Fig. 15:** Distribution of the backward processes along the time. DFT of the kernel associated with the distribution of the backward processes generated by the different algorithms at different times for the first motion blur, for the first fabric texture in Figure 8. Observe that the distributions of IIIGDM and CGDM are well aligned with the theoretical true conditional distribution near the final times, including  $t = 50$ .



**Fig. 16: Observation of the bias for the different algorithms for the SR problem.** Illustration of the algorithms' bias for the SR problems with  $r = 8$  and  $\sigma = 10/255$ . As observed in [16], the theoretical mean is noised at this level of noise. The observations are similar to the deblurring case.

## References

- [1] Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Foundations and Trends in Machine Learning **12**(4), 307–392 (2019) <https://doi.org/10.1561/2200000056>
- [2] Prost, J., Houdard, A., Almansa, A., Papadakis, N.: Diverse super-resolution with pretrained deep hierarchical vaes. In: 19th International Joint Conference on Computer Vision Theory and Applications. VISAPP2024 (2024)
- [3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020) <https://doi.org/10.1145/3422622>
- [4] Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **43**(12), 4217–4228 (2021) <https://doi.org/10.1109/TPAMI.2020.2970919>
- [5] Lugmayr, A., Danelljan, M., Van Gool, L., Timofte, R.: SRFlow: Learning the super-resolution space with normalizing flow. In: SRFlow, pp. 715–732 (2020)
- [6] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS ’21 (2021)
- [7] Daras, G., Chung, H., Lai, C.-H., Mitsufuji, Y., Milanfar, P., Dimakis, A.G., Ye, C., Delbracio, M.: A survey on diffusion models for inverse problems (2024)
- [8] Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. In: The Eleventh International Conference on Learning Representations (2023)
- [9] Chung, H., Sim, B., Ye, J.C.: Improving diffusion models for inverse problems using manifold constraints. (2022)
- [10] Song, J., Vahdat, A., Mardani, M., Kautz, J.: Pseudoinverse-guided diffusion models for inverse problems. In: International Conference on Learning Representations (2023)
- [11] Kawar, B., Elad, M., Ermon, S., Song, J.: Denoising diffusion restoration models. In: ICLR Workshop on Deep Generative Models for Highly Structured Data (2022)

- [12] Moroy, L., Bourmaud, G., Champagnat, F., Giovannelli, J.-F.: Evaluating the Posterior Sampling Ability of Plug&Play Diffusion Methods in Sparse-View CT (2025). <https://arxiv.org/abs/2410.21301>
- [13] Rozet, F., Andry, G., Lanusse, F., Louuppe, G.: Learning diffusion priors from observations by expectation maximization. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
- [14] Thong, D.Y.W., Mbakam, C.K., Pereyra, M.: Do Bayesian imaging methods report trustworthy probabilities? (2024). <https://arxiv.org/abs/2405.08179>
- [15] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems (2017)
- [16] Pierret, E., Galerne, B.: Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. In: Forty-second International Conference on Machine Learning (2025)
- [17] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations. In: 9th International Conference on Learning Representations, ICLR (2021)
- [18] Strasman, S., Ocello, A., Boyer, C., Corff, S.L., Lemaire, V.: An analysis of the noise schedule for score-based generative models. Transactions on Machine Learning Research (2025)
- [19] Hurault, S., Terris, M., Moreau, T., Peyré, G.: From Score Matching to Diffusion: A Fine-Grained Error Analysis in the Gaussian Setting (2025). <https://arxiv.org/abs/2503.11615>
- [20] Galerne, B., Gousseau, Y., Morel, J.-M.: Random Phase Textures: Theory and Synthesis. IEEE Transactions on Image Processing **20**(1), 257–267 (2011)
- [21] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020 (2020)
- [22] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of the 32nd International Conference on Machine Learning (2015)
- [23] Manzano, E., Gomez-Villegas, M., Marin, J.: A matrix variate generalization of the power exponential family of distributions. Communication in Statistics- Theory and Methods (2002)
- [24] Pascal, F., Bombrun, L., Tourneret, J.-Y., Berthoumieu, Y.: Parameter estimation for multivariate generalized gaussian distributions. IEEE Transactions on Signal Processing (2013)
- [25] Pierret, E., Galerne, B.: Stochastic super-resolution for Gaussian microtextures. SIAM Journal on Imaging Sciences **18**(2), 1176–1207 (2025)
- [26] Peyré, G., Cuturi, M., (2019). <https://doi.org/10.1561/2200000073>
- [27] Xia, G.-S., Ferradans, S., Peyré, G., Aujol, J.-F.: Synthesizing and mixing stationary Gaussian texture models. SIAM Journal on Imaging Sciences **7**(1), 476–508 (2014) <https://doi.org/10.1137/130920562>
- [28] Galerne, B., Leclaire, A.: Texture Inpainting Using Efficient Gaussian Conditional Simulation. SIAM Journal on Imaging Sciences (2017)
- [29] Bishop, C.M.: Pattern Recognition and Machine Learning. Information science and statistics, vol. 4. Springer, ??? (2006)
- [30] Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics, ???

(2002)

- [31] Golub, G.H., Loan, C.F.: Matrix Computations, 4th edn. JHU Press, ??? (2013)

## Appendix A Closed-form expressions for Gaussian diffusion models

In this appendix, we provide closed-form expressions for Gaussian distributions in the context of diffusion models. We first present a lemma to compute conditional Gaussian distributions and gives a compact expression for the denoised estimate  $\hat{x}_0(\mathbf{x}_t)$  and use the first lemma to compute the theoretical  $p(\mathbf{x}_t | \mathbf{v})$ ,  $p(\mathbf{v} | \mathbf{x}_t)$  and the theoretical discrete backward process, which sheds light on the inexactness of CGDM as observed in the Wasserstein error plots. Finally, we provide the computation of  $p_t(\mathbf{x}_t | \mathbf{v})$  given the noisy likelihood modeled by the different algorithms which is used to study them in a forward form in Section 3.2.

### A.1 General computation of conditional Gaussian distributions

In the following, we derive a lemma to compute conditional Gaussian distributions in our context. A first idea was to use the lemma from Section 2.3.3 in [29] but it needs invertibility assumption on the covariance matrices. For this reason, we propose a more general kriging reasoning providing the following lemma.

**Lemma 1** (Conditional Gaussian distribution computations using a kriging reasoning). *Given the assumptions*

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{m}, \mathbf{\Gamma}) \quad (\text{A1})$$

$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{Bx}, \tau^2 \mathbf{I}) \quad (\text{A2})$$

$$(\text{A3})$$

the conditional distribution of  $\mathbf{x}$  given  $\mathbf{y}$  is given by

$$p(\mathbf{x} | \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}) \quad (\text{A4})$$

with

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}} = \mathbf{m} + \mathbf{\Gamma} \mathbf{B}^T \mathbf{M}^{-1} (\mathbf{v} - \mathbf{Bm}) \quad (\text{A5})$$

$$\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}} = \mathbf{\Gamma} - \mathbf{\Gamma} \mathbf{B}^T \mathbf{M}^{-1} \mathbf{B}\mathbf{\Gamma} \quad (\text{A6})$$

$$\mathbf{M} = \mathbf{B}\mathbf{\Gamma}\mathbf{B}^T + \tau^2 \mathbf{I}. \quad (\text{A7})$$

Note that  $\mathbf{M}$  is invertible because  $\mathbf{B}\mathbf{\Gamma}\mathbf{B}^T$  is a positive symmetric matrix.

*Proof* In the case  $\mathbf{m} = \mathbf{0}$ , as shown by a kriging reasoning in Appendix E of [25], by denoting  $\mathbf{M} = \mathbf{B}\mathbf{\Gamma}\mathbf{B}^T + \tau^2 \mathbf{I}$ ,

$$\boldsymbol{\Lambda}^T \mathbf{y} + \tilde{\mathbf{x}} - \boldsymbol{\Lambda}^T (\mathbf{B}\tilde{\mathbf{x}} + \tau \tilde{\mathbf{n}}) \quad (\text{A8})$$

where  $\tilde{\mathbf{x}}$  is a sample from  $p_0$  independent of  $\mathbf{y}$  and  $\tilde{\mathbf{n}}$  is an independent sample following  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  follows the posterior distribution  $p(\mathbf{x} | \mathbf{y})$  with  $\boldsymbol{\Lambda} = \mathbf{M}^{-1} \mathbf{B}\mathbf{\Gamma}$  is solution of a kriging equation. Consequently, the posterior covariance matrix is the covariance matrix of this expression with respect to  $\mathbf{x}$  which is

$$\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}} = (\mathbf{I} - \boldsymbol{\Lambda}^T \mathbf{B}) \mathbf{\Gamma} (\mathbf{I} - \boldsymbol{\Lambda}^T \mathbf{B})^T + \tau^2 \boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \quad (\text{A9})$$

$$= \mathbf{\Gamma} - \boldsymbol{\Lambda}^T \mathbf{B}\mathbf{\Gamma} + \boldsymbol{\Lambda}^T \mathbf{B}\mathbf{\Gamma}\mathbf{B}^T \boldsymbol{\Lambda}^T - \mathbf{\Gamma} \mathbf{B}^T \boldsymbol{\Lambda}^T + \tau^2 \boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \quad (\text{A10})$$

$$= \mathbf{\Gamma} - \boldsymbol{\Lambda}^T \mathbf{B}\mathbf{\Gamma} + \boldsymbol{\Lambda}^T (\mathbf{B}\mathbf{\Gamma}\mathbf{B}^T + \mathbf{\Gamma}\mathbf{B}^T + \tau^2 \mathbf{I}) \boldsymbol{\Lambda} - \mathbf{\Gamma} \mathbf{B}^T \boldsymbol{\Lambda}^T \quad (\text{A11})$$

$$= \mathbf{\Gamma} - \boldsymbol{\Lambda}^T \mathbf{B}\mathbf{\Gamma} + \boldsymbol{\Lambda}^T \mathbf{B}\mathbf{\Gamma} - \mathbf{\Gamma} \mathbf{B}^T \boldsymbol{\Lambda}^T \quad (\text{A12})$$

$$= \mathbf{\Gamma} - \mathbf{\Gamma} \mathbf{B}^T \mathbf{M}^{-1} \mathbf{B}\mathbf{\Gamma} \quad (\text{A13})$$

and the conditional expectation

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}} = \boldsymbol{\Gamma}\mathbf{B}^T\mathbf{M}^{-1}\mathbf{y}. \quad (\text{A14})$$

In the general case  $\mathbf{m} \neq \mathbf{0}$ , the covariance matrix is not affected and the conditional expectation becomes

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}} = \boldsymbol{\mu} + \boldsymbol{\Gamma}\mathbf{B}^T\mathbf{M}^{-1}(\mathbf{y} - \mathbf{B}\mathbf{m}). \quad (\text{A15})$$

which is equivalent to rescale the quantity to ensure a null expectation.  $\square$

## A.2 Computation of $\hat{\mathbf{x}}_0(\mathbf{x}_t)$

In the Gaussian setting  $p_0 = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , the expression for  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$  is given by Tweedie's formula (see Equation (15)), which, when combined with the explicit form of the score function (see Equation (29)), yields the closed-form expression

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t + (1 - \bar{\alpha}_t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)) \quad (\text{A16})$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}) \right) \quad (\text{A17})$$

$$= (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu} + \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\mathbf{x}_t) \quad (\text{A18})$$

$$= (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu} + \frac{\boldsymbol{\Sigma}_t^{-1}}{\sqrt{\bar{\alpha}_t}} (\boldsymbol{\Sigma}_t\mathbf{x}_t - (1 - \bar{\alpha}_t)\mathbf{x}_t) \quad (\text{A19})$$

$$= (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu} + \frac{\boldsymbol{\Sigma}_t^{-1}}{\sqrt{\bar{\alpha}_t}} (\bar{\alpha}_t\boldsymbol{\Sigma}\mathbf{x}_t) \quad (\text{A20})$$

$$= (1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu} + \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}\mathbf{x}_t. \quad (\text{A21})$$

Then, one can also derive

$$(1 - \bar{\alpha}_t)\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu} + \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}\mathbf{x}_t = \boldsymbol{\Sigma}_t^{-1}((1 - \bar{\alpha}_t)\mathbf{I})\boldsymbol{\mu} + \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}\mathbf{x}_t \quad (\text{A22})$$

$$= \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\Sigma}_t - \bar{\alpha}_t\boldsymbol{\Sigma})\boldsymbol{\mu} + \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}\mathbf{x}_t \quad (\text{A23})$$

$$= \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}). \quad (\text{A24})$$

## A.3 Computation of $p(\mathbf{x}_t | \mathbf{v})$

In this section, we compute  $p(\mathbf{x}_0 | \mathbf{v})$  and  $p(\mathbf{x}_t | \mathbf{v})$  in the Gaussian setting  $p_0 = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . We apply Lemma 1 with

$$\mathbf{x} = \mathbf{x}_0 \quad (\text{A25})$$

$$\mathbf{y} = \mathbf{v} \quad (\text{A26})$$

$$\mathbf{m} = \boldsymbol{\mu} \quad (\text{A27})$$

$$\boldsymbol{\Gamma} = \boldsymbol{\Sigma} \quad (\text{A28})$$

$$\mathbf{B} = \mathbf{A} \quad (\text{A29})$$

$$\tau = \sigma. \quad (\text{A30})$$

By denoting  $\mathbf{M} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A} + \sigma^2\mathbf{I}$ , it ensures that

$$p(\mathbf{x}_0 | \mathbf{v}) = \mathcal{N}(\boldsymbol{\mu}_{0|\mathbf{v}}, \boldsymbol{\Sigma}_{0|\mathbf{v}}) \quad (\text{A31})$$

with

$$\boldsymbol{\Sigma}_{\mathbf{x}_0|\mathbf{v}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^{-1} \mathbf{A} \boldsymbol{\Sigma} \quad (\text{A32})$$

$$\boldsymbol{\mu}_{0|\mathbf{v}} = \boldsymbol{\mu} + \boldsymbol{\Sigma} \mathbf{A}^T \mathbf{M}^{-1} (\mathbf{v} - \mathbf{A} \boldsymbol{\mu}) \quad (\text{A33})$$

$$\mathbf{M} = \mathbf{A} \boldsymbol{\Sigma} \mathbf{A}^T + \sigma^2 \mathbf{I}. \quad (\text{A34})$$

Then, because  $p(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$ ,

$$p(\mathbf{x}_t \mid \mathbf{v}) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \mathbb{E}[\mathbf{x}_0 \mid \mathbf{v}], \bar{\alpha}_t \boldsymbol{\Sigma}_{\mathbf{x}_0|\mathbf{v}} + (1 - \bar{\alpha}_t) \mathbf{I}) \quad (\text{A35})$$

#### A.4 Computation of $p(\mathbf{v} \mid \mathbf{x}_t)$

Let us first compute  $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ . We apply Lemma 1 with

$$\mathbf{x} = \mathbf{x}_0 \quad (\text{A36})$$

$$\mathbf{y} = \mathbf{x}_t \quad (\text{A37})$$

$$\mathbf{m} = \boldsymbol{\mu} \quad (\text{A38})$$

$$\boldsymbol{\Gamma} = \boldsymbol{\Sigma} \quad (\text{A39})$$

$$\mathbf{B} = \sqrt{\bar{\alpha}_t} \mathbf{I} \quad (\text{A40})$$

$$\tau = (1 - \bar{\alpha}_t). \quad (\text{A41})$$

We obtain the covariance matrix is

$$\boldsymbol{\Sigma}_{0|t} = \boldsymbol{\Sigma} - \bar{\alpha}_t \boldsymbol{\Sigma} (\bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I})^{-1} \boldsymbol{\Sigma} = \boldsymbol{\Sigma} - \bar{\alpha}_t \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \quad (\text{A42})$$

$$= \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} (\boldsymbol{\Sigma}_t - \bar{\alpha}_t \boldsymbol{\Sigma}) \quad (\text{A43})$$

$$= (1 - \bar{\alpha}_t) \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1}, \quad (\text{A44})$$

and the expectation is  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ .

Then,  $\mathbf{v} = \mathbf{A} \mathbf{x}_0 + \sigma^2 \mathbf{I}$  and consequently,

$$p(\mathbf{v} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{A} \hat{\mathbf{x}}_0(\mathbf{x}_t), (1 - \bar{\alpha}_t) \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T + \sigma^2 \mathbf{I}). \quad (\text{A45})$$

#### A.5 Computation of the theoretical backward $p(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$

In this section, we prove that the theoretical backward process associated with diffusion models applied to Gaussian distributions does not, in general, have a diagonal covariance matrix. We apply Lemma 1 with

$$\mathbf{x} = \mathbf{x}_{t-1} \quad (\text{A46})$$

$$\mathbf{y} = \mathbf{x}_t \quad (\text{A47})$$

$$\mathbf{m} = \sqrt{\bar{\alpha}_{t-1}} \boldsymbol{\mu} \quad (\text{A48})$$

$$\boldsymbol{\Gamma} = \boldsymbol{\Sigma}_{t-1} \quad (\text{A49})$$

$$\mathbf{B} = \sqrt{\alpha_t} \mathbf{I} \quad (\text{A50})$$

$$\tau = \sqrt{\beta_t}. \quad (\text{A51})$$

With

$$\mathbf{M} = \alpha_t \boldsymbol{\Sigma}_{t-1} + \beta_t \mathbf{I} \quad (\text{A52})$$

$$= \alpha_t (\bar{\alpha}_{t-1} \boldsymbol{\Sigma} + (1 - \bar{\alpha}_{t-1}) \mathbf{I}) + (1 - \alpha_t) \mathbf{I} \quad (\text{A53})$$

$$= \bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I} \quad (\text{A54})$$

$$= \boldsymbol{\Sigma}_t \quad (\text{A55})$$

The covariance matrix of the distribution  $p(\mathbf{x}_{t-1} | \mathbf{x}_t)$  is given by

$$\boldsymbol{\Sigma}_{t-1|t} = \boldsymbol{\Sigma}_{t-1} - \alpha_t \boldsymbol{\Sigma}_{t-1}^2 \boldsymbol{\Sigma}_t^{-1} \quad (\text{A56})$$

$$= \alpha_t \boldsymbol{\Sigma}_{t-1}^2 \boldsymbol{\Sigma}_t^{-1} (\boldsymbol{\Sigma}_t - \alpha_t \boldsymbol{\Sigma}_{t-1}) = \boldsymbol{\Sigma}_{t-1} \boldsymbol{\Sigma}_t^{-1} (\bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I} - \bar{\alpha}_t \boldsymbol{\Sigma} - (\alpha_t - \bar{\alpha}_t) \mathbf{I}) \quad (\text{A57})$$

$$= \beta_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{\Sigma}_t^{-1} \quad (\text{A58})$$

and the conditional expectation is

$$\mathbb{E}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \sqrt{\bar{\alpha}_t} \boldsymbol{\mu} + \sqrt{\alpha_t} \boldsymbol{\Sigma}_t^{-1} \boldsymbol{\Sigma}_{t-1} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}). \quad (\text{A59})$$

Using the identity

$$\boldsymbol{\Sigma}_t = \bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I} = \alpha_t \boldsymbol{\Sigma}_{t-1} + \beta_t \mathbf{I} \quad (\text{A60})$$

$$\iff \boldsymbol{\Sigma}_{t-1} = \frac{1}{\alpha_t} (\boldsymbol{\Sigma}_t - \beta_t \mathbf{I}), \quad (\text{A61})$$

we obtain

$$\mathbb{E}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}} \left( (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}) - \beta_t \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}) \right) + \sqrt{\bar{\alpha}_{t-1}} \boldsymbol{\mu} \quad (\text{A62})$$

$$= \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \beta_t \boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}) \right) \quad (\text{A63})$$

$$= \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t + \beta_t \nabla \log p_t(\mathbf{x}_t)) \quad (\text{A64})$$

which is the correct expression for the conditional expectation given by the backward process (see Equation (5)).

## A.6 Computation of $p(\mathbf{x}_t | \mathbf{v})$ for the different algorithms

In the Gaussian setting where  $p_0 = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , the forward process defined in Equation (1) yields  $p_t(\mathbf{x}_t) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \boldsymbol{\mu}, \boldsymbol{\Sigma}_t)$  with  $\boldsymbol{\Sigma}_t = \bar{\alpha}_t \boldsymbol{\Sigma} + (1 - \bar{\alpha}_t) \mathbf{I}$  and  $p(\mathbf{v} | \mathbf{x}_t) = \mathcal{N}(\mathbf{A}\hat{\mathbf{x}}_0(\mathbf{x}_t), \mathbf{C}_{\mathbf{v}|t})$ , as developed in Section 2.2. By Bayes' rule,

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t | \mathbf{v}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}} \log p_t(\mathbf{v} | \mathbf{x}_t) \quad (\text{A65})$$

Consequently,

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t | \mathbf{v}) = -\boldsymbol{\Sigma}_t^{-1} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}) - \sqrt{\bar{\alpha}_t} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \mathbf{C}_{\mathbf{v}|t}^{-1} (\mathbf{A}\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{v}) \quad (\text{A66})$$

By denoting  $p_t(\mathbf{x}_t | \mathbf{v}) = \mathcal{N}(\boldsymbol{\mu}_{t|\mathbf{v}}, \mathbf{C}_{t|\mathbf{v}})$ ,  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t | \mathbf{v}) = -\mathbf{C}_{\mathbf{v}|t}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{\mathbf{v}|\mathbf{x}_t})$  and by identifying the terms in  $\mathbf{x}_t$ ,

$$\mathbf{C}_{t|\mathbf{v}}^{-1} = \boldsymbol{\Sigma}_t^{-1} + \bar{\alpha}_t \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \mathbf{A}^T \mathbf{C}_{\mathbf{v}|t}^{-1} \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\Sigma}_t^{-1} \quad (\text{A67})$$

Furthermore, by Woodbury matrix identity [30],

$$(B + UDV)^{-1} = B^{-1} - B^{-1}U(D^{-1} + VB^{-1}U)^{-1}VB^{-1} \quad (\text{A68})$$

Consequently, with  $B = \Sigma_t, U = -\sqrt{\bar{\alpha}_t} \mathbf{A} \Sigma^T, V = \sqrt{\bar{\alpha}_t} \mathbf{A} \Sigma$ , and  $D$  such that  $D^{-1} + VB^{-1}U = \mathbf{C}_{v|t}$  ie  $D = (\mathbf{C}_{v|t} + \bar{\alpha}_t \mathbf{A} \Sigma^2 \Sigma_t^{-1} \mathbf{A}^T)^{-1}$ ,

$$\mathbf{C}_{t|v} = \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \mathbf{C}_{v|t} - \bar{\alpha}_t \mathbf{A} \Sigma^2 \Sigma_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A69})$$

and finally,

$$\mathbf{C}_{t|v}^{\text{DPS}} = \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \bar{\alpha}_t \mathbf{A} \Sigma^2 \Sigma_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A70})$$

$$\mathbf{C}_{t|v}^{\text{IGDM}} = \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \sigma^2 \mathbf{I} + (1 - \bar{\alpha}_t) \mathbf{A} \mathbf{A}^T + \bar{\alpha}_t \mathbf{A} \Sigma^2 \Sigma_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A71})$$

$$\mathbf{C}_{t|v}^{\text{CGDM}} = \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \sigma^2 \mathbf{I} + (1 - \bar{\alpha}_t) \mathbf{A} \Sigma \Sigma_t^{-1} \mathbf{A}^T + \bar{\alpha}_t \mathbf{A} \Sigma^2 \Sigma_t^{-1} \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A72})$$

$$= \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \mathbf{A} \Sigma \Sigma_t^{-1} ((1 - \bar{\alpha}_t) \mathbf{I} + \bar{\alpha}_t \Sigma) \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A73})$$

$$= \Sigma_t - \bar{\alpha}_t \Sigma \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \mathbf{A} \Sigma \mathbf{A}^T \right)^{-1} \mathbf{A} \Sigma \quad (\text{A74})$$

By identifying the other terms,

$$\mathbf{C}_{t|v}^{-1} \boldsymbol{\mu}_{t|v} = \sqrt{\bar{\alpha}_t} \Sigma_t^{-1} \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \Sigma \Sigma_t^{-1} \mathbf{A}^T \mathbf{C}_{v|t}^{-1} (\mathbf{v} + \bar{\alpha}_t \mathbf{A} \Sigma \Sigma_t^{-1} \boldsymbol{\mu} - \mathbf{A} \boldsymbol{\mu}) \quad (\text{A75})$$

$$= \sqrt{\bar{\alpha}_t} \left( \Sigma_t^{-1} + \bar{\alpha}_t \Sigma \Sigma_t^{-1} \mathbf{A}^T \mathbf{C}_{v|t}^{-1} \mathbf{A} \Sigma \Sigma_t^{-1} \right) \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \Sigma \Sigma_t^{-1} \mathbf{A}^T \mathbf{C}_{v|t}^{-1} (\mathbf{v} - \mathbf{A} \boldsymbol{\mu}) \quad (\text{A76})$$

$$= \sqrt{\bar{\alpha}_t} \mathbf{C}_{t|v}^{-1} \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \Sigma \Sigma_t^{-1} \mathbf{A}^T \mathbf{C}_{v|t}^{-1} (\mathbf{v} - \mathbf{A} \boldsymbol{\mu}), \quad (\text{A77})$$

and

$$\boldsymbol{\mu}_{t|v} = \sqrt{\bar{\alpha}_t} \boldsymbol{\mu} + \sqrt{\bar{\alpha}_t} \mathbf{C}_{t|v} \Sigma \Sigma_t^{-1} \mathbf{A}^T \mathbf{C}_{v|t}^{-1} (\mathbf{v} - \mathbf{A} \boldsymbol{\mu}). \quad (\text{A78})$$

## Appendix B Analytical derivations for diffusion models on Gaussian microtextures

### B.1 Application of the exact score function

The ADSN model [20] allows for the exact computation of the score function associated with the iterations of diffusion models. The inverse of the matrix  $\Sigma_t = \bar{\alpha}_t \Sigma + (1 - \bar{\alpha}_t) \mathbf{I}$  appears in the score function, as given in Equation (29). Let us now describe why this inversion is feasible.

First, we recall Equation (57): for  $\xi \in \Omega_{M,N}$ ,

$$\widehat{\Sigma}_t(\xi) = \bar{\alpha}_t \widehat{\mathbf{t}}(\xi) \left[ \widehat{\mathbf{t}}(\xi) \right]^T + (1 - \bar{\alpha}_t) \mathbf{I}_3. \quad (\text{57})$$

In a certain sense, the action of  $\Sigma_t$  is separable across all frequencies. We can invert it using the following lemma.

**Lemma 2.** *Let  $\mathbf{y} \in \mathbb{C}^3, a, b \in \mathbb{R}^+$ ,*

$$(a\mathbf{y}\mathbf{y}^T + b\mathbf{I}_3)^{-1} = \frac{1}{b} \mathbf{I}_3 - \frac{a\|\mathbf{y}\|^2}{b(a\|\mathbf{y}\|^2 + b)} \mathbf{y}\mathbf{y}^T. \quad (\text{B79})$$

*Proof* It is well-known that  $\frac{\mathbf{y}\bar{\mathbf{y}}^T}{\|\mathbf{y}\|^2}$  is the orthogonal projection on  $\text{span}(\mathbf{y})$  (see for example [31]). Consequently, by completing  $\frac{\mathbf{y}}{\|\mathbf{y}\|^2}$  in an orthogonal basis and considering its matrix  $\mathbf{P}$ ,

$$\mathbf{y}\bar{\mathbf{y}}^T = \mathbf{P}^T \begin{pmatrix} \|\mathbf{y}\|^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{P}. \quad (\text{B80})$$

Then,

$$\mathbf{P}(a\mathbf{y}\bar{\mathbf{y}}^T + bI_3) = \mathbf{P}^T \begin{pmatrix} a\|\mathbf{y}\|^2 + b & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{pmatrix} \mathbf{P} \quad (\text{B81})$$

and

$$(a\mathbf{y}\bar{\mathbf{y}}^T + bI_3)^{-1} = \mathbf{P}^T \begin{pmatrix} \frac{1}{a\|\mathbf{y}\|^2 + b} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ 0 & 0 & \frac{1}{b} \end{pmatrix} \mathbf{P} \quad (\text{B82})$$

$$= \frac{1}{b} I_3 + \left( \frac{1}{a\|\mathbf{y}\|^2 + b} - \frac{1}{b} \right) \mathbf{y}\bar{\mathbf{y}}^T \quad (\text{B83})$$

$$= \frac{1}{b} I_3 - \frac{a\|\mathbf{y}\|^2}{b(a\|\mathbf{y}\|^2 + b)}. \quad (\text{B84})$$

□

This lemma follows from the fact that the symmetric rank-one matrix  $\frac{\mathbf{y}\bar{\mathbf{y}}^T}{\|\mathbf{y}\|^2}$  is the orthogonal projection onto  $\text{span}(\mathbf{y})$ . It can be applied to the matrix  $\Sigma_t$  separately at each frequency, following Equation (57). To be complete, for  $\mathbf{x} \in \mathbb{R}^{3\Omega_{M,N}}$  and  $\xi \in \Omega_{M,N}$ ,

$$\widehat{\Sigma}_t^{-1} \mathbf{x}(\xi) = \frac{1}{1 - \bar{\alpha}_t} \widehat{\mathbf{x}}(\xi) - \frac{\bar{\alpha}_t \|\widehat{\mathbf{t}}(\xi)\|^2}{(1 - \bar{\alpha}_t)(\bar{\alpha}_t \|\widehat{\mathbf{t}}(\xi)\|^2 + \bar{\alpha}_t)} \widehat{\Sigma}(\xi) \widehat{\mathbf{x}}(\xi) \quad (\text{B85})$$

where  $\widehat{\mathbf{x}}(\xi) = (\widehat{\mathbf{x}}_1(\xi) \ \widehat{\mathbf{x}}_2(\xi) \ \widehat{\mathbf{x}}_3(\xi))^T \in \mathbb{R}^3$

## B.2 Proof of Proposition 1

We recall here Proposition 1 that we prove.

**Proposition 1** (Simultaneous diagonalizability of the Gaussian backward processes associated with the different algorithms). *For the deblurring problem involving ADSN microtextures, the covariance matrices associated with the backward processes of the different algorithms— $(\Sigma_t^{CGDM})_{0 \leq t \leq T}$ ,  $(\Sigma_t^{DPS})_{0 \leq t \leq T}$ , and  $(\Sigma_t^{\Pi GDM})_{0 \leq t \leq T}$ —are diagonalizable in the same orthogonal basis as  $\Sigma$ .*

*Proof* As proved in Appendix I.2 of [16], it is sufficient to study the 3D structure of the covariance matrix of ADSN to determine its eigenvectors and associated eigenvalues. In a certain sense, the matrix is block-diagonalizable with respect to 3D blocks. We will show that its eigenvectors are preserved over time. For a given frequency  $\xi \in \Omega_{M,N}$ , we denote

$$\widehat{\mathbf{v}}_1(\xi) = \widehat{\mathbf{t}}(\xi) = \begin{pmatrix} \widehat{\mathbf{t}}_1(\xi) \\ \widehat{\mathbf{t}}_2(\xi) \\ \widehat{\mathbf{t}}_3(\xi) \end{pmatrix}, \widehat{\mathbf{v}}_2(\xi) = \begin{pmatrix} -\bar{\widehat{\mathbf{t}}}_3(\xi) \\ 0 \\ \bar{\widehat{\mathbf{t}}}_1(\xi) \end{pmatrix}, \widehat{\mathbf{v}}_3(\xi) = \begin{pmatrix} 0 \\ -\bar{\widehat{\mathbf{t}}}_3(\xi) \\ \bar{\widehat{\mathbf{t}}}_2(\xi) \end{pmatrix}. \quad (\text{B86})$$

It is an orthogonal basis of eigenvectors of  $\widehat{\Sigma}(\xi)$  because

$$\bar{\widehat{\mathbf{v}}}_k(\xi)^T \widehat{\mathbf{v}}_\ell(\xi) = 0 \text{ for } 1 \leq k < \ell \leq 3 \quad (\text{B87})$$

$$\widehat{\Sigma}(\xi) \widehat{\mathbf{v}}_1(\xi) = [\bar{\widehat{\mathbf{t}}}(\xi)^T \widehat{\mathbf{t}}(\xi)] \widehat{\mathbf{v}}_1(\xi) \quad (\text{B88})$$

$$\widehat{\Sigma}(\xi) \widehat{\mathbf{v}}_2(\xi) = \mathbf{0} \quad (\text{B89})$$

$$\widehat{\Sigma}(\xi)\widehat{v}_3(\xi) = \mathbf{0}. \quad (\text{B90})$$

We denote  $\lambda_k(\xi)$  the eigenvalues associated with  $\widehat{v}_k(\xi)$  (respectively  $[\bar{\mathbf{t}}(\xi)^T \bar{\mathbf{t}}(\xi)]$ , 0 and 0). Furthermore,  $\widehat{\mathbf{C}}(\xi)$  conserves the same eigenvectors because

$$\widehat{\mathbf{C}}(\xi)\widehat{v}_k(\xi) = \mathbf{c}(\xi)\widehat{v}_k(\xi) \text{ for } 1 \leq k \leq 3 \quad (\text{B91})$$

and  $\widehat{\Sigma}_t^{-1}(\xi)$  also by the previous subsection. This ensures that the matrix  $\mathbf{A}_t^{\text{algo}}$  given in Equation (48) and recalled here

$$\mathbf{A}_t^{\text{algo}} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{I} - \beta_t \Sigma_t^{-1} - \beta_t \bar{\alpha}_t \Sigma_t^{-1} \Sigma \mathbf{A}^T \left( \mathbf{C}_{v|t}^{\text{algo}} \right)^{-1} \mathbf{A} \Sigma \Sigma_t^{-1} \right) \quad (\text{B92})$$

preserves the eigenvectors of  $\widehat{\Sigma}(\xi)$ . Finally, the application of Algorithm 5 ensures that the covariance matrices associated with the algorithms' backward processes preserve the eigenvector basis.

**Remark 4.** *Equation (B91) shows that our method cannot be extended to cases where different kernels are applied to different image channels, highlighting the crucial role of the specific structure of the deblurring problem degradation operator.*

□