On the Accuracy of Diffusion Models in Bayesian Image Inverse Problems: A Gaussian Case Study

Émile Pierret[a], supervised by Bruno Galerne[a,b]

French Biennal of Applied and Industrial Mathematics (SMAI), June 6, 2025

[a] Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS
[b] Institut universitaire de France (IUF)

Introduction

Inverse problems

$$v = Ax + \sigma n, \quad n \sim \mathcal{N}_0 \tag{1}$$

where A is an inpainting, super-resolution, or blur operator.

Clean image | Super-resolution $(\times 4)$ | Inpainting | Gaussian blur | Motion blur

Inverse problems

$$v = Ax + \sigma n, \quad n \sim \mathcal{N}_0 \tag{1}$$

where A is an inpainting, super-resolution, or blur operator.

Clean image | Super-resolution ($\times 4$) | Inpainting | Gaussian blur | Motion blur

$\Longrightarrow$ One popular solution: diffusion models.

Ground truth	Degraded image	DDRM [Kawar et al., 2022]	DPS [Chung et al., 2023]	ΠGDM [Song et al., 2023]

Ground truth	Degraded image	DDRM [Kawar et al., 2022]	DPS [Chung et al., 2023]	ΠGDM [Song et al., 2023]

Ground truth | Degraded image | DDRM [Kawar et al., 2022] | DPS [Chung et al., 2023] | ΠGDM [Song et al., 2023]

Ground truth	Degraded image	DDRM [Kawar et al., 2022]	DPS [Chung et al., 2023]	ΠGDM [Song et al., 2023]

One pending question

$$v = Ax + \sigma n, \quad n \sim \mathcal{N}_0 \tag{2}$$

where A inpainting, SR or blur operator.

[1] E.P & Galerne, B. (2025). Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. *Forty-second International Conference on Machine Learning*

One pending question

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x} + \sigma \boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}_0 \tag{2}$$

where $\boldsymbol{A}$ inpainting, SR or blur operator.

Aim: Sample from $p(\boldsymbol{x} \mid \boldsymbol{v})$

[1] E.P & Galerne, B. (2025). Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. *Forty-second International Conference on Machine Learning*

One pending question

$$v = Ax + \sigma n, \quad n \sim \mathcal{N}_0 \tag{2}$$

where A inpainting, SR or blur operator.

Aim: Sample from $p(x \mid v)$

Question: To what extent do diffusion models allow sampling from the target posterior distribution?

[1] E.P & Galerne, B. (2025). Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. *Forty-second International Conference on Machine Learning*

One pending question

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x} + \sigma \boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}_0 \tag{2}$$

where $\boldsymbol{A}$ inpainting, SR or blur operator.

Aim: Sample from $p(\boldsymbol{x} \mid \boldsymbol{v})$

Question: To what extent do diffusion models allow sampling from the target posterior distribution?

Idea: Observe what happens for Gaussian image distributions, that allow for feasible calculations.
(follows the previous work [E.P and Galerne, 2025][1])

[1] E.P & Galerne, B. (2025). Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. *Forty-second International Conference on Machine Learning*

Diffusion models for image generation

Discrete DDPM [Ho et al., 2020][2]

Forward process

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_t, \quad 1 \leqslant t \leqslant T, \quad \boldsymbol{z}_t \sim \mathcal{N}_0, \quad \boldsymbol{x}_0 \sim p_{\text{data}}, \tag{3}$$

Ones can write

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}_0 \tag{4}$$

[2] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*

Forward process

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_t, \quad 1 \leqslant t \leqslant T, \quad \boldsymbol{z}_t \sim \mathcal{N}_0, \quad \boldsymbol{x}_0 \sim p_{\text{data}}, \tag{3}$$

Ones can write

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}_0 \tag{4}$$

Backward process

By learning $\boldsymbol{\varepsilon}_\theta$ such that $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \approx \varepsilon_t$,

$$\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\boldsymbol{y}_t, t)\right) + \sqrt{\tilde{\beta}_t}\boldsymbol{z}_t, \quad \boldsymbol{z}_t \sim \mathcal{N}_0. \tag{5}$$

[2] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*

Discrete DDPM [Ho et al., 2020][2]

Forward process

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_t, \quad 1 \leqslant t \leqslant T, \quad z_t \sim \mathcal{N}_0, \quad x_0 \sim p_{\text{data}}, \tag{3}$$

Ones can write

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}_0 \tag{4}$$

Backward process

By learning $\boldsymbol{\varepsilon}_\theta$ such that $\boldsymbol{\varepsilon}_\theta(x_t, t) \approx \varepsilon_t$,

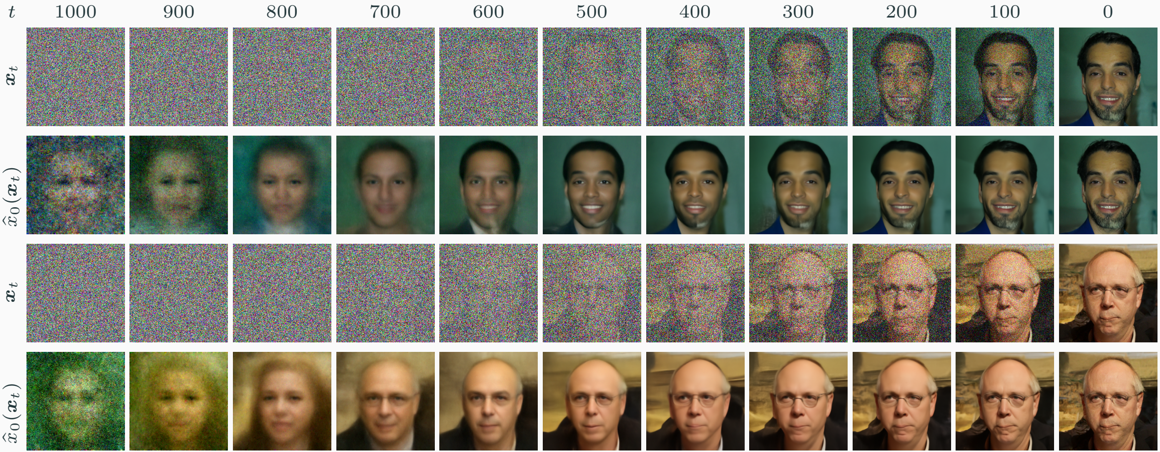
$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(y_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(y_t, t)\right) + \sqrt{\tilde{\beta}_t}z_t, \quad z_t \sim \mathcal{N}_0. \tag{5}$$

By denoting p_t the marginals of the forward process and learning $s_\theta(x, t) \approx \nabla_x \log p_t(x_t)$,

$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(y_t + \beta_t s_{\theta^\star}(y_t, t)) + \sigma_t z_t, z_t \sim \mathcal{N}_0, 1 \leqslant t \leqslant T, z_t \sim \mathcal{N}_0, y_T \sim \mathcal{N}_0. \tag{6}$$

[2] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*

Generation examples



Diffusion models for inverse problems

And for solving image problems ?

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma\boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{7}$$

Aim: Sampling $p_0(\cdot \mid \boldsymbol{v})$

And for solving image problems ?

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma\boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{7}$$

Aim: Sampling $p_0(\cdot \mid \boldsymbol{v})$

$\Longrightarrow$ **conditional forward process**

$$\tilde{\boldsymbol{x}}_t = \sqrt{1-\beta_t}\tilde{\boldsymbol{x}}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_t, \quad 1 \leqslant t \leqslant T, \quad \boldsymbol{z}_t \sim \mathcal{N}_0, \quad \tilde{\boldsymbol{x}}_0 \sim p_{\text{data}}(\cdot \mid \boldsymbol{v}), \tag{8}$$

And for solving image problems ?

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{7}$$

Aim: Sampling $p_0(\cdot \mid \boldsymbol{v})$

$\Longrightarrow$ **conditional forward process**

$$\tilde{\boldsymbol{x}}_t = \sqrt{1-\beta_t}\tilde{\boldsymbol{x}}_{t-1} + \sqrt{\beta_t}\boldsymbol{z}_t, \quad 1 \leqslant t \leqslant T, \quad \boldsymbol{z}_t \sim \mathcal{N}_0, \quad \tilde{\boldsymbol{x}}_0 \sim p_{\text{data}}(\cdot \mid \boldsymbol{v}), \tag{8}$$

$\Longrightarrow$ **conditional backward process**

$$\tilde{\boldsymbol{y}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\tilde{\boldsymbol{y}}_t + \beta_t \nabla \log \tilde{p}_t(\tilde{\boldsymbol{y}}_t)\right) + \sigma_t \boldsymbol{z}_t, \quad \boldsymbol{z}_t \sim \mathcal{N}_0, 1 \leqslant t \leqslant T, \boldsymbol{z}_t \sim \mathcal{N}_0, \tilde{\boldsymbol{y}}_T \sim \mathcal{N}_0 \tag{9}$$

Bayes theorem

Our point of interest is $\nabla \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla \log p_t(\boldsymbol{x}_t \mid \boldsymbol{v})$.

Bayes theorem

Our point of interest is $\nabla \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla \log p_t(\boldsymbol{x}_t \mid \boldsymbol{v})$. By Bayes' formula,

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{10}$$

Bayes theorem

Our point of interest is $\nabla \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla \log p_t(\boldsymbol{x}_t \mid \boldsymbol{v})$. By Bayes' formula,

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{10}$$

Algorithm 5
Unconditional DDPM backward process

$\boldsymbol{y}_T \sim \mathcal{N}_0$
for t=T **to** 1 **do**
  $\boldsymbol{z}_t \sim \mathcal{N}_0$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t + \beta_t s_{\theta^\star}(\boldsymbol{y}_t, t)\right) + \sigma_t \boldsymbol{z}_t$
end for

Algorithm 6 Conditional DDPM backward process

$\boldsymbol{y}_T \sim \mathcal{N}_0$
for $t = T$ **to** 1 **do**
  $\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(\boldsymbol{x}_t + (1-\overline{\alpha}_t) s_{\theta^\star}(\boldsymbol{y}_t, t)\right)$
  $\tilde{s}_\theta(\boldsymbol{y}_t, t) = s_{\theta^\star}(\boldsymbol{y}_t, t) + \nabla \log p_t(\boldsymbol{x} \mid \boldsymbol{v})$
  $\boldsymbol{z}_t \sim \mathcal{N}_0$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t + \beta_t \tilde{s}_{\theta^\star}(\boldsymbol{y}_t, t)\right) + \sigma_t \boldsymbol{z}_t$
end for

Description of two algorithms from the literature

Mean of $p_t(v \mid x_t)$

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{11}$$

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{12}$$

Some assumptions lead to "$p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is Gaussian".

Mean of $p_t(v \mid x_t)$

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma\boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{11}$$

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{12}$$

Some assumptions lead to "$p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is Gaussian". Let note that an approximation of $\mathbb{E}(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is known.

By Tweedie's formula, that is,

$$\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) := \mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right] = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(\boldsymbol{x}_t + (1 - \overline{\alpha}_t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right). \tag{13}$$

Mean of $p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{11}$$

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{12}$$

Some assumptions lead to "$p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is Gaussian". Let note that an approximation of $\mathbb{E}(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is known.

By Tweedie's formula, that is,

$$\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) := \mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right] = \frac{1}{\sqrt{\overline{\alpha}_t}} \left(\boldsymbol{x}_t + (1 - \overline{\alpha}_t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right). \tag{13}$$

As a consequence, $\mathbb{E}\left[\boldsymbol{v} \mid \boldsymbol{x}_t\right]$ is given by

$$\mathbb{E}\left[\boldsymbol{v} \mid \boldsymbol{x}_t\right] = \boldsymbol{A}\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \frac{1}{\sqrt{\overline{\alpha}_t}} \boldsymbol{A}\left(\boldsymbol{x}_t + (1 - \overline{\alpha}_t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right). \tag{14}$$

Mean of $p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}, \quad \boldsymbol{x}_0 \sim p_0, \boldsymbol{n} \sim \mathcal{N}_0 \tag{11}$$

$$\nabla_{\boldsymbol{x}} \log \tilde{p}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{12}$$

Some assumptions lead to "$p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is Gaussian". Let note that an approximation of $\mathbb{E}(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is known.

By Tweedie's formula, that is,

$$\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) := \mathbb{E}\left[\boldsymbol{x}_0 \mid \boldsymbol{x}_t\right] = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(\boldsymbol{x}_t + (1 - \overline{\alpha}_t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right). \tag{13}$$

As a consequence, $\mathbb{E}\left[\boldsymbol{v} \mid \boldsymbol{x}_t\right]$ is given by

$$\mathbb{E}\left[\boldsymbol{v} \mid \boldsymbol{x}_t\right] = \boldsymbol{A}\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \frac{1}{\sqrt{\overline{\alpha}_t}}\boldsymbol{A}\left(\boldsymbol{x}_t + (1 - \overline{\alpha}_t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right). \tag{14}$$

Finally,

$$p(\boldsymbol{x}_t \mid \boldsymbol{v}) = \mathcal{N}(\boldsymbol{A}\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \boldsymbol{C}_{\boldsymbol{v}|t}) \tag{15}$$

with $\boldsymbol{C}_{\boldsymbol{v}|t}$ to fix.

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Denoising Posterior Sampling (DPS) algorithm [Chung et al., 2023][3]

$$\log p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \tag{16}$$

[3] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Denoising Posterior Sampling (DPS) algorithm [Chung et al., 2023][3]

$$\log p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \tag{16}$$

$$p(\boldsymbol{v} \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{A}\boldsymbol{x}_0, \sigma^2 \boldsymbol{I}\right) \text{ (by } \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}\text{).} \tag{17}$$

[3] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Denoising Posterior Sampling (DPS) algorithm [Chung et al., 2023][3]

$$\log p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \tag{16}$$

$$p(\boldsymbol{v} \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{A}\boldsymbol{x}_0, \sigma^2 \boldsymbol{I}\right) \text{ (by } \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}\text{).} \tag{17}$$

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) = -\frac{1}{2\sigma^2} \nabla_{\boldsymbol{x}_t} \left\| \boldsymbol{v} - \boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t) \right\|^2. \tag{18}$$

[3] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Denoising Posterior Sampling (DPS) algorithm [Chung et al., 2023][3]

$$\log p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \tag{16}$$

$$p(\boldsymbol{v} \mid \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{A}\boldsymbol{x}_0, \sigma^2 \boldsymbol{I}\right) \text{ (by } \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}_0 + \sigma \boldsymbol{n}\text{).} \tag{17}$$

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) = -\frac{1}{2\sigma^2} \nabla_{\boldsymbol{x}_t} \left\| \boldsymbol{v} - \boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t) \right\|^2 . \tag{18}$$

In practice,

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \approx -\frac{\alpha_{\text{DPS}}}{2\sigma^2} \nabla_{\boldsymbol{x}_t} \left\| \boldsymbol{v} - \boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t) \right\|^2 . \tag{19}$$

$\Longrightarrow \boldsymbol{C}_{\boldsymbol{v}|t}^{\text{DPS}} = \frac{\sigma^2}{\alpha_{\text{DPS}}} \boldsymbol{I}$

[3] Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Pseudo-Guided Diffusion model (ΠGDM) algorithm [Song et al., 2023][4]

$$p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) \approx \mathcal{N}\left(\widehat{x}_0(\boldsymbol{x}_t), r_t^2 \boldsymbol{I}\right). \tag{20}$$

[4] Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. *International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Pseudo-Guided Diffusion model (ΠGDM) algorithm [Song et al., 2023][4]

$$p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) \approx \mathcal{N}\left(\widehat{x}_0(\boldsymbol{x}_t), r_t^2 \boldsymbol{I}\right). \tag{20}$$

  Consequently,

$$p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \mathcal{N}\left(\boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t), r_t^2 \boldsymbol{A}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}\right) \tag{21}$$

[4] Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. *International Conference on Learning Representations*

Covariance of $p_t(v \mid x)$

We denote it $\boldsymbol{C}_{\boldsymbol{v}|t}$.

- Pseudo-Guided Diffusion model (ΠGDM) algorithm [Song et al., 2023][4]

$$p(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) \approx \mathcal{N}\left(\widehat{x}_0(\boldsymbol{x}_t), r_t^2 \boldsymbol{I}\right). \tag{20}$$

Consequently,

$$p(\boldsymbol{v} \mid \boldsymbol{x}_t) \approx \mathcal{N}\left(\boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t), r_t^2 \boldsymbol{A}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}\right) \tag{21}$$

$\Longrightarrow \boldsymbol{C}_{\boldsymbol{v}|t}^{\Pi\text{GDM}} = r_t^2 \boldsymbol{A}\boldsymbol{A}^T + \sigma^2 \boldsymbol{I}$

[4] Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. *International Conference on Learning Representations*

Under Gaussian assumption

By adding the assumption: $\boldsymbol{x}_0$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Under Gaussian assumption

By adding the assumption: $\boldsymbol{x}_0$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

$$p(\boldsymbol{v} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{A}\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t), (1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}\right), \tag{22}$$

$$\text{with } \widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \boldsymbol{\mu} + \sqrt{\overline{\alpha}_t}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}(\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{\mu}). \tag{23}$$

We call this setting "Conditional Gaussian Diffusion Model" (CGDM) $\Longrightarrow \boldsymbol{C}_{\boldsymbol{v}|t}^{\mathsf{CGDM}} = (1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}$

Comparison of the choice made by different algorithms

	$\boldsymbol{C}_{\boldsymbol{v}\|t}$ (Covariance of $p(\boldsymbol{v} \mid \boldsymbol{x}_t)$)
DPS Chung et al., 2023	$\frac{\sigma^2}{\alpha_{\text{DPS}}}\boldsymbol{I}$
ΠGDM Song et al., 2023	$(1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}$
CGDM	$(1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}, \quad \boldsymbol{\Sigma}_t = \overline{\alpha}_t\boldsymbol{\Sigma} + (1-\overline{\alpha}_t)\boldsymbol{I}$

Comparison of the choice made by different algorithms

	$\boldsymbol{C}_{\boldsymbol{v}\mid t}$ (Covariance of $p(\boldsymbol{v} \mid \boldsymbol{x}_t)$)
DPS Chung et al., 2023	$\frac{\sigma^2}{\alpha_{\text{DPS}}}\boldsymbol{I}$
ΠGDM Song et al., 2023	$(1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}$
CGDM	$(1-\overline{\alpha}_t)\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_t^{-1}\boldsymbol{A}^T + \sigma^2\boldsymbol{I}, \quad \boldsymbol{\Sigma}_t = \overline{\alpha}_t\boldsymbol{\Sigma} + (1-\overline{\alpha}_t)\boldsymbol{I}$

Algorithm 7

Unconditional DDPM backward process

$\boldsymbol{y}_T \sim \mathcal{N}_0$
for t=T **to** 1 **do**
  $\boldsymbol{z}_t \sim \mathcal{N}_0$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t + \beta_t s_{\theta^\star}(\boldsymbol{y}_t, t)\right) + \sigma_t \boldsymbol{z}_t$
end for

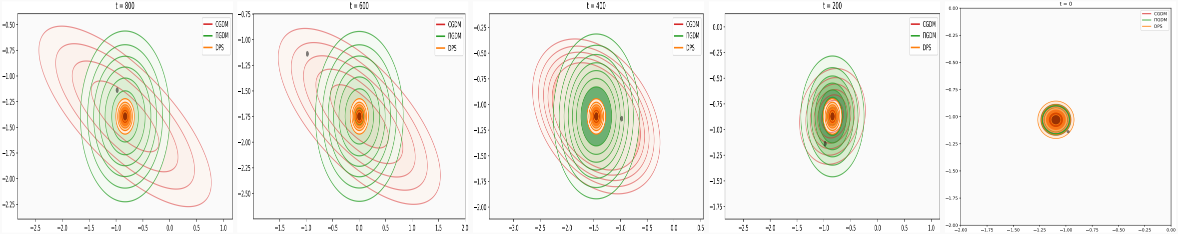
Algorithm 8 Conditional DDPM backward process

$\boldsymbol{y}_T \sim \mathcal{N}_0$
for $t = T$ **to** 1 **do**
  $\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(\boldsymbol{x}_t + (1-\overline{\alpha}_t)s_{\theta^\star}(\boldsymbol{y}_t, t)\right)$
  $\tilde{s}_\theta(\boldsymbol{y}_t, t) = s_{\theta^\star}(\boldsymbol{y}_t, t) - \frac{1}{2}\nabla_{\boldsymbol{x}_t}\left\|\boldsymbol{v} - \boldsymbol{A}\widehat{\boldsymbol{x}}_0(\boldsymbol{x}_t)\right\|^2_{\boldsymbol{C}^{-1}_{\boldsymbol{v}\mid t}}$
  $\boldsymbol{z}_t \sim \mathcal{N}_0$
  $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t + \beta_t \tilde{s}_{\theta^\star}(\boldsymbol{y}_t, t)\right) + \sigma_t \boldsymbol{z}_t$
end for

Comparison of the choice made by different algorithms

	$C_{v\|t}$ (Covariance of $p(v \mid x_t)$)
DPS Chung et al., 2023	$\frac{\sigma^2}{\alpha_{\text{DPS}}} I$
ΠGDM Song et al., 2023	$(1 - \overline{\alpha}_t) A A^T + \sigma^2 I$
CGDM	$(1 - \overline{\alpha}_t) A \Sigma \Sigma_t^{-1} A^T + \sigma^2 I, \quad \Sigma_t = \overline{\alpha}_t \Sigma + (1 - \overline{\alpha}_t) I$

$t = 800$	600	400	200	0



At $t \approx 0$, $p(v \mid x_0) = \mathcal{N}(A x_0, \sigma^2 I)$

Comparison of the algorithms via 2-Wasserstein distance

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$$ **which is a stationary law**

$\boldsymbol{u}$ Samples of $\mathrm{ADSN}(\boldsymbol{u})$

Image extracted from [Galerne et al., 2011a][5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\gamma = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

$\boldsymbol{u}$ Samples of $\mathrm{ADSN}(\boldsymbol{u})$

Image extracted from [Galerne et al., 2011a][5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\gamma = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

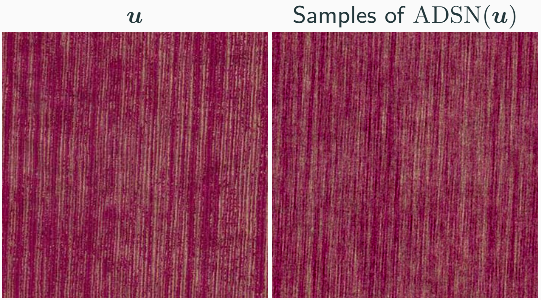


Image extracted from [Galerne et al., 2011a][5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathscr{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\boldsymbol{\gamma} = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

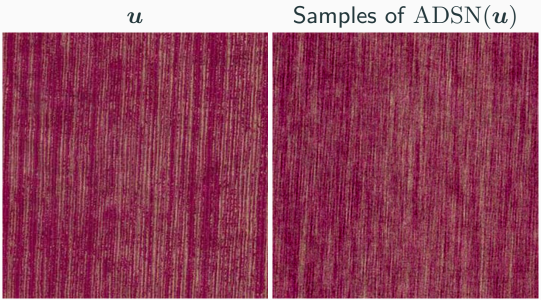


Image extracted from [Galerne et al., 2011a] [5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\boldsymbol{\gamma} = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

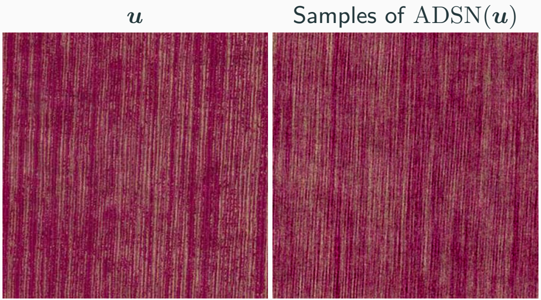


Image extracted from [Galerne et al., 2011a] [5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$$ **which is a stationary law**

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\boldsymbol{\gamma} = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

$\boldsymbol{u}$ | Samples of $\mathrm{ADSN}(\boldsymbol{u})$

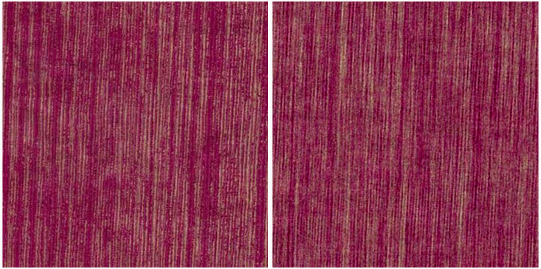


Image extracted from [Galerne et al., 2011a][5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne et al., 2011b] [6]

Let $\boldsymbol{u} \in \mathbb{R}^{\Omega_{M,N}}$ be a grayscale image, m its grayscale mean and $\boldsymbol{t} = \frac{1}{\sqrt{MN}}(\boldsymbol{u} - m)$ its associated texton. Let $\boldsymbol{w}$ be a white Gaussian noise,

$$\boldsymbol{X} = \boldsymbol{t} \star \boldsymbol{w} \sim \mathrm{ADSN}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \quad \textbf{which is a stationary law}$$

$\boldsymbol{\Gamma}$ represents the convolution by the kernel $\gamma = \boldsymbol{t} \star \check{\boldsymbol{t}}$.

$\boldsymbol{u}$ | Samples of $\mathrm{ADSN}(\boldsymbol{u})$

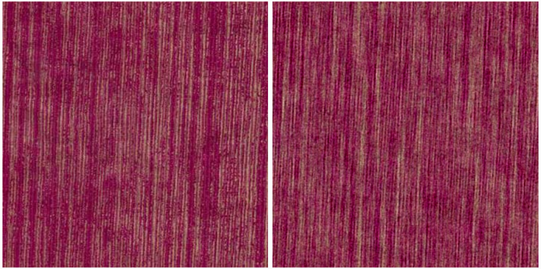


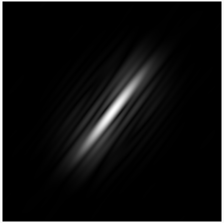
Image extracted from [Galerne et al., 2011a][5]

[5] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*

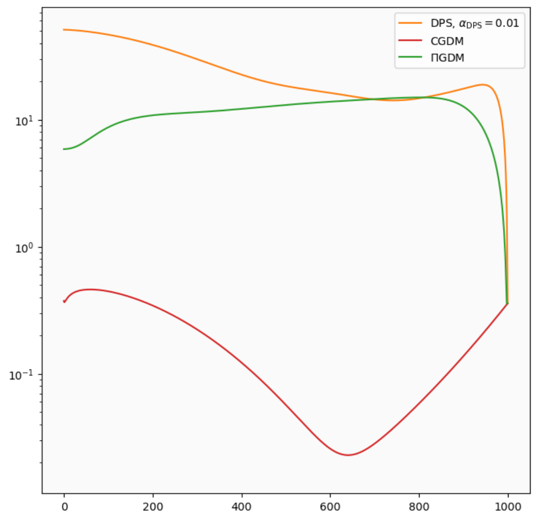
[6] Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267

Exact Wasserstein error for deblurring

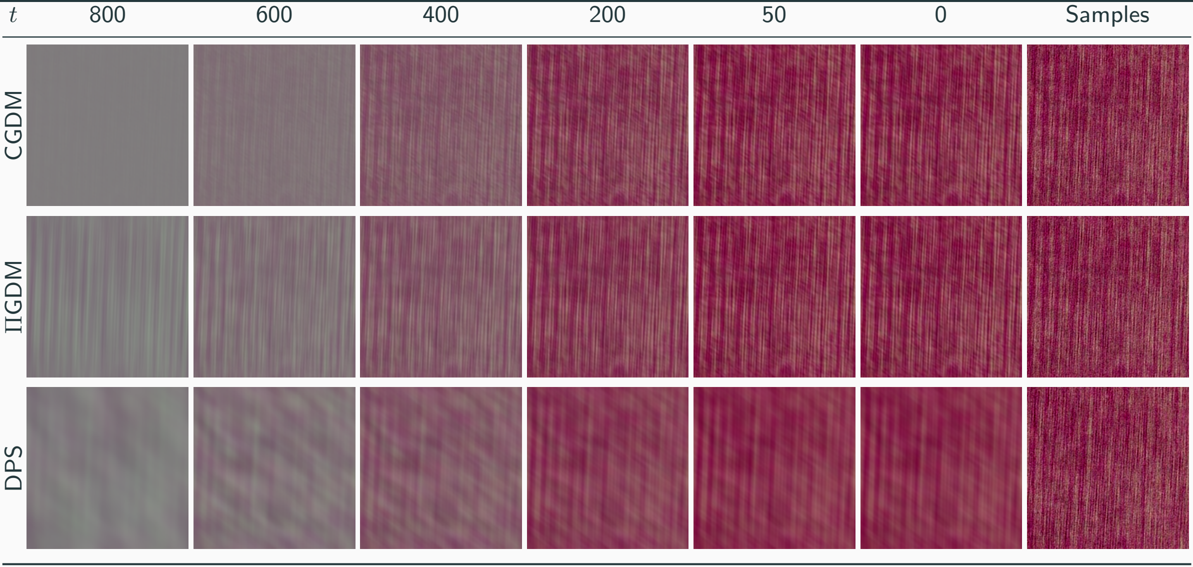
Blur kernel



Blurred image v



Study of the bias



Reverse Bayes rule

We use

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{v}) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{24}$$

where $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$ is known and $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is modelized (with a covariance matrix $\boldsymbol{C}_{\boldsymbol{v}|t}$).

Reverse Bayes rule

We use

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{v}) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t), \tag{24}$$

where $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$ is known and $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{v} \mid \boldsymbol{x}_t)$ is modelized (with a covariance matrix $\boldsymbol{C}_{\boldsymbol{v}|t}$). From these expressions, we can write $p_t(\boldsymbol{x}_t \mid \boldsymbol{v}) = \mathcal{N}(\boldsymbol{\mu}_{t|\boldsymbol{v}}, \boldsymbol{C}_{t|\boldsymbol{v}})$ with

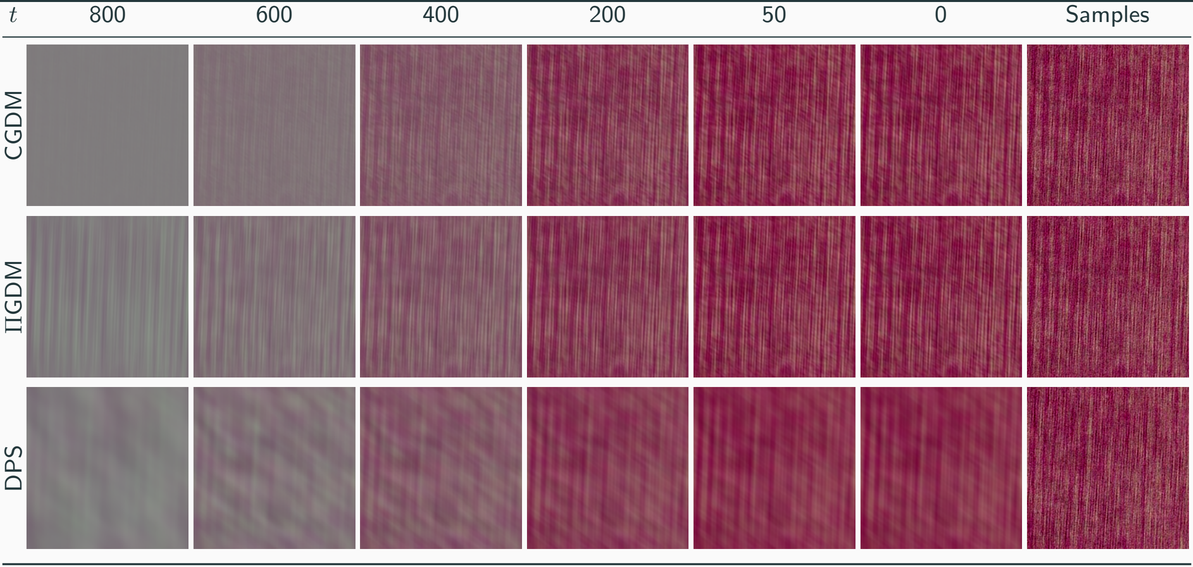
$$\boldsymbol{C}_{t|\boldsymbol{v}}^{\text{DPS}} = \overline{\alpha}_t \left[\boldsymbol{\Sigma} - \boldsymbol{\Sigma} \boldsymbol{A}^T \left(\frac{\sigma^2}{\alpha_{\text{DPS}}} \boldsymbol{I} + \overline{\alpha}_t \boldsymbol{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \boldsymbol{A}^T \right)^{-1} \boldsymbol{A} \boldsymbol{\Sigma} \right] + (1 - \overline{\alpha}_t) \boldsymbol{I} \tag{25}$$

$$\boldsymbol{C}_{t|\boldsymbol{v}}^{\Pi\text{GDM}} = \overline{\alpha}_t \left[\boldsymbol{\Sigma} - \boldsymbol{\Sigma} \boldsymbol{A}^T \left(\sigma^2 \boldsymbol{I} + (1 - \overline{\alpha}_t) \boldsymbol{A} \boldsymbol{A}^T + \overline{\alpha}_t \boldsymbol{A} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}_t^{-1} \boldsymbol{A}^T \right)^{-1} \boldsymbol{A} \boldsymbol{\Sigma} \right] + (1 - \overline{\alpha}_t) \boldsymbol{I} \tag{26}$$

$$\boldsymbol{C}_{t|\boldsymbol{v}}^{\text{CGDM}} = \overline{\alpha}_t \left[\boldsymbol{\Sigma} - \boldsymbol{\Sigma} \boldsymbol{A}^T (\sigma^2 \boldsymbol{I} + \boldsymbol{A} \boldsymbol{\Sigma} \boldsymbol{A}^T)^{-1} \boldsymbol{A} \boldsymbol{\Sigma} \right] + (1 - \overline{\alpha}_t) \boldsymbol{I} \tag{27}$$

At $t \approx 0$, $\boldsymbol{C}_{t|\boldsymbol{v}}^{\text{DPS}} \approx \boldsymbol{C}_{t|\boldsymbol{v}}^{\Pi\text{GDM}} \approx \boldsymbol{C}_{t|\boldsymbol{v}}^{\text{CGDM}}$.

Study of the bias



Conclusion

Conclusion

- Study of the DPS in practice,

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \approx -\frac{\alpha_{\mathsf{DPS}}}{2\sigma^2} \nabla_{\boldsymbol{x}_t} \left\| \boldsymbol{v} - \boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t) \right\|^{1}. \tag{28}$$

- Generalization to other inverse problems.
- Generalization to multimodal distributions.
- An important direction of research [Rozet et al., 2024][7]:

$$\mathrm{Cov}(\boldsymbol{x} \mid \boldsymbol{x}_t) = \sigma_t^2 + \sigma_t^4 \nabla_{\boldsymbol{x}}^2 \log p_t(\boldsymbol{x}_t), \tag{29}$$

[7] Rozet, F., Andry, G., Lanusse, F., & Louppe, G. (2024). Learning diffusion priors from observations by expectation maximization. *The Thirty-eighth Annual Conference on Neural Information Processing Systems*

Conclusion

- Study of the DPS in practice,

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{v} \mid \boldsymbol{x}_0 = \widehat{x}_0(\boldsymbol{x}_t)) \approx -\frac{\alpha_{\mathsf{DPS}}}{2\sigma^2} \nabla_{\boldsymbol{x}_t} \|\boldsymbol{v} - \boldsymbol{A}\widehat{x}_0(\boldsymbol{x}_t)\|^{1} . \tag{28}$$

- Generalization to other inverse problems.
- Generalization to multimodal distributions.
- An important direction of research [Rozet et al., 2024][7]:

$$\mathrm{Cov}(\boldsymbol{x} \mid \boldsymbol{x}_t) = \sigma_t^2 + \sigma_t^4 \nabla_{\boldsymbol{x}}^2 \log p_t(\boldsymbol{x}_t), \tag{29}$$

Thank you for your attention !

[7] Rozet, F., Andry, G., Lanusse, F., & Louppe, G. (2024). Learning diffusion priors from observations by expectation maximization. *The Thirty-eighth Annual Conference on Neural Information Processing Systems*

References

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. *The Eleventh International Conference on Learning Representations*.

E.P & Galerne, B. (2025). Diffusion models for gaussian distributions: Exact solutions and wasserstein errors. *Forty-second International Conference on Machine Learning*.

Galerne, B., Gousseau, Y., & Morel, J.-M. (2011a). Micro-texture synthesis by phase randomization. *Image Processing On Line*.

Galerne, B., Gousseau, Y., & Morel, J.-M. (2011b). Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Kawar, B., Elad, M., Ermon, S., & Song, J. (2022). Denoising diffusion restoration models. *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

Bibliography ii

Rozet, F., Andry, G., Lanusse, F., & Louppe, G. (2024). Learning diffusion priors from observations by expectation maximization. *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. *International Conference on Learning Representations.*