# Monte Carlo sampling and deep generative models for Bayesian inference

Florentin Coeurdoux, PhD Student, *Toulouse University*

November 23th, 2023

# Bayesian inference

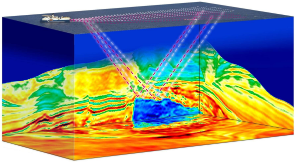
$\mathbf{x}$: unknown object of interest
$\mathbf{y}$: available data (observation)

Update of prior information with available data[1]:

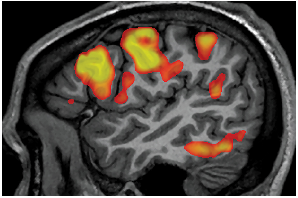
$$\begin{array}{ccccc} \text{Prior} & \times & \text{Likelihood} & \rightarrow & \text{Posterior} \\ p(\mathbf{x}) & & p(\mathbf{y} \mid \mathbf{x}) & & p(\mathbf{x} \mid \mathbf{y}) \\ \uparrow & & \uparrow & & \uparrow \\ \text{pre-observation} & & \text{acquisition} & & \text{best} \\ \text{knowledge} & & \text{process} & & \text{guess} \end{array}$$

[1] For complete review, see *Robert and Casella, 1999*

# Various applications

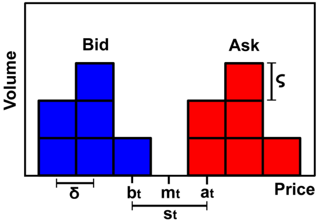


(a) Seismic imaging



(b) Brain fMRI

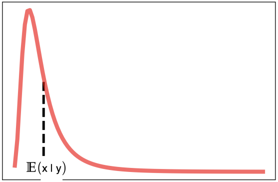
(c) Earth observation




(d) LOB analysis

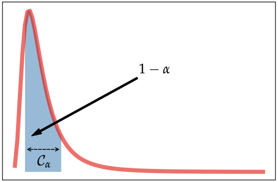
# Bayesian inference : Uncertainty quantification

Bayesian estimators




$$\arg\min \int h(\mathbf{x}) p(\mathbf{x} \mid \mathbf{y}) \mathrm{d}\mathbf{x}$$

Credibility regions $\mathcal{C}_\alpha$




$$\int_{\mathcal{C}_\alpha} p(\mathbf{x} \mid \mathbf{y}) d\mathbf{x} = 1-\alpha, \quad \alpha \in (0,1)$$

$\rightarrow$ Computing such integrals is not straightforward

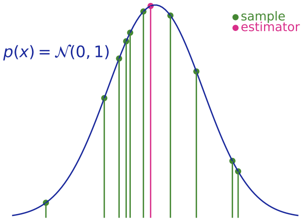
# Monte Carlo integration

One alternative lies in making use of sampling to approximate integration

$$\int h(\mathbf{x})p(\mathbf{x} \mid \mathbf{y})d\mathbf{x} \approx \frac{1}{N}\sum_{n=1}^{N} h\left(\mathbf{x}^{(n)}\right), \quad \mathbf{x}^{(n)} \sim p(\mathbf{x} \mid \mathbf{y})$$

Sampling challenges :

- high-dimensional: $\mathbf{x} \in \mathbb{R}^d$ with $d \gg 1$
- composite: $-\log p(\mathbf{x} \mid \mathbf{y}) = \sum_{i=1}^{b} U_i(\mathbf{x})$
- complicated: non-smooth/conjugate $\cdots$



# Sampling: Model based vs Data driven

## Starting hypothesis

What information about $p(\mathbf{x})$ is available ?

- model based, unnormalized density $p(\mathbf{x}) \propto \exp[-U(\mathbf{x})]$
- data driven, samples $\{\mathbf{x}_i\}_{i=0}^N \sim p(\mathbf{x})$

| Field | Model based | Data driven |
|---|---|---|
| Information | $p(\mathbf{x}) \propto \exp[-U(\mathbf{x})]$ | $\{\mathbf{x}_i\}_{i=0}^N \sim p(\mathbf{x})$ |
| Methods | Monte Carlo | Learn mapping $f_\sharp q_Z \approx p_X$ |

# A myriad of Monte Carlo methods

Metropolis-Hastings

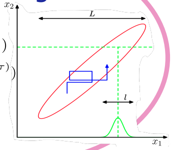
$$\min\left(1, \frac{p(x')}{p(x)} \frac{g(x|x')}{g(x'|x)}\right)$$

Gibbs sampling

$$p(x_1 \mid x_2^{(\tau)}, x_3^{(\tau)})$$

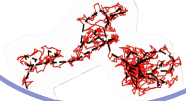
$$p(x_2 \mid x_1^{(\tau+1)}, x_3^{(\tau)})$$

$$\dots$$



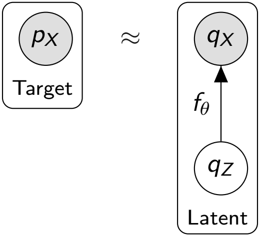

Langevin sampling

$$dX_t = \nabla_x \log p\left(X_t\right) dt + \sigma_t dB_t$$

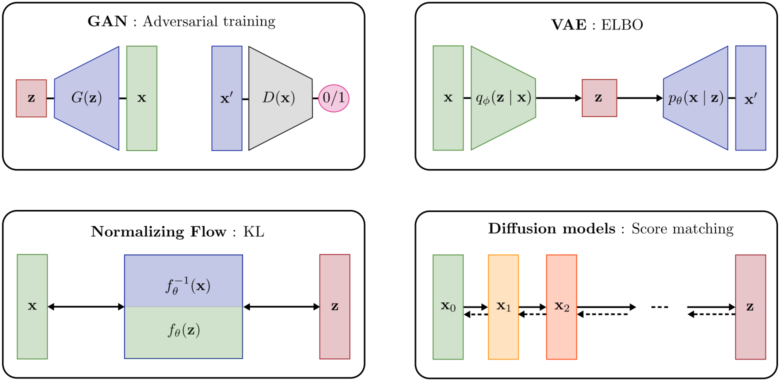


# What is deep generative modeling ?

- Aim: Approximate distribution $q_X \approx p_X$ such that $q_X := f_{\theta\sharp} q_Z$
- A general approach :
  - Samples $\{x_i\}_{i=0}^{N}$ from $p_X$ defined over $\mathcal{X}$.
  - Choose a simple latent distribution $q_Z$ defined over $\mathcal{Z}$.
  - Learn a mapping $f_\theta : \mathcal{Z} \to \mathcal{X}$ by minimizing $D_{\mathrm{Stat}}(p_X, f_{\theta\sharp} q_Z)$ over $\theta$.
- Ancestral sampling : Latent $\mathbf{z} \sim q_Z \Rightarrow$ Target $q_X \sim \hat{\mathbf{x}} = f(\mathbf{z})$



# A myriad of models



# Contributions

**Submitted journal articles**

- Coeurdoux et al. "*Normalizing flow sampling with Langevin dynamics in the latent space*", 2nd round JMLR
- Coeurdoux et al. "*Plug-and-Play split Gibbs sampler: embedding deep generative priors in Bayesian inference*", IEEE Transactions on Image Processing

**International conferences**

- Coeurdoux et al. "*Sliced-Wasserstein normalizing flows: beyond maximum likelihood training*". ESANN 2022
- Coeurdoux et al. "*Learning Optimal Transport Between two Empirical Distributions with Normalizing Flows*". ECLM-PKDD 2022

**National conferences**

- Coeurdoux et al. "*Méthode MCMC plug-and-play avec a priori génératif profond*". GRETSI 2023
- Coeurdoux et al. "*Approximation du transport optimal entre distributions empiriques par flux de normalisation*". GRETSI 2022

**Talk**

- Workshop: Geostat Days, "*Solving Inverse Problem with deep learning*", Mines Paris PSL, Sept 2023
- Seminar: SIOP seminar, "*Split Gibbs Plug-and-Play Sampler: Diffusion Models for inverse problem*", University of Bordeaux, May 2023.
- Workshop: Interfacing Bayesian statistics and machine learning, "*Langevin based Normalizing flow sampling*", Bayes Centre, Edinburgh, Jan 2023
- Seminar: D2 Reading Group, "*Normalizing flow sampling with Langevin dynamics in the latent space*", Oxford University, Dec 2022

...

# Contributions

**Submitted journal articles**

- Coeurdoux et al. *"Normalizing flow sampling with Langevin dynamics in the latent space"*, 2nd round JMLR
- Coeurdoux et al. *"Plug-and-Play split Gibbs sampler: embedding deep generative priors in Bayesian inference"*, IEEE Transactions on Image Processing

**International conferences**

- Coeurdoux et al. *"Sliced-Wasserstein normalizing flows: beyond maximum likelihood training"*. ESANN 2022
- Coeurdoux et al. *"Learning Optimal Transport Between two Empirical Distributions with Normalizing Flows"*. ECLM-PKDD 2022

**National conferences**

- Coeurdoux et al. *"Méthode MCMC plug-and-play avec a priori génératif profond"*. GRETSI 2023
- Coeurdoux et al. *"Approximation du transport optimal entre distributions empiriques par flux de normalisation"*. GRETSI 2022

**Talk**

- Workshop: Geostat Days, *"Solving Inverse Problem with deep learning"*, Mines Paris PSL, Sept 2023
- Seminar: SIOP seminar, *"Split Gibbs Plug-and-Play Sampler: Diffusion Models for inverse problem"*, University of Bordeaux, May 2023.
- Workshop: Interfacing Bayesian statistics and machine learning, *"Langevin based Normalizing flow sampling"*, Bayes Centre, Edinburgh, Jan 2023
- Seminar: D2 Reading Group, *"Normalizing flow sampling with Langevin dynamics in the latent space"*, Oxford University, Dec 2022

...

# Plug-and-Play split Gibbs sampler: Motivation

**Inverse Problem**

$$\underset{\text{Observation}}{\mathbf{y}} \approxeq \underset{\text{Data acquisition}}{\mathcal{A}(\mathbf{x})}$$

- **solve complex** ill-posed ML or inverse problems
- **big** data in high dimensions
- **good** performance
- **fast** inference algorithms
- **credibility intervals**

# The usual toolbox of inference

- **Optimization**:
  - problem $\Rightarrow$ loss function
  - efficient algorithms
  - theoretical guarantees
  - interpretability / functional analysis
- **Bayesian approaches**:
  - probabilitic models
  - uncertainty quantification
- **Machine learning** (deep):
  - adaptive $\Rightarrow$ relevant
  - outstanding performance

**Toward the best of all worlds?**

# The optimization-based approach

Inverse problem $\Rightarrow$ **cost function**

$$\mathbf{y} \approxeq \mathcal{A}(\mathbf{x}) \quad \Rightarrow \quad \hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) + g(\mathbf{x})$$

⚠ Explicit regularizations are significantly outperformed by DNN

- Half quadratic splitting (HQS)

$$L_\rho^{(\mathrm{HQS})}(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}, \mathbf{y}) + g(\mathbf{z}) + \frac{1}{2\rho^2}\|\mathbf{x} - \mathbf{z}\|_2^2,$$

- Decomposes into simpler sub problems:

$$\mathbf{x} \leftarrow \mathrm{prox}_f(\mathbf{z}) = \arg\min_{\{\mathbf{x}\}} L_\rho^{(\mathrm{HQS})}(\mathbf{x}, \mathbf{z}) = \arg\min_{\{\mathbf{x}\}} f(\mathbf{x}, \mathbf{y}) + \frac{1}{2\rho^2}\|\mathbf{x} - \mathbf{z}\|_2^2$$

$$\mathbf{z} \leftarrow \mathrm{prox}_g(\mathbf{x}) = \arg\min_{\{\mathbf{z}\}} L_\rho^{(\mathrm{HQS})}(\mathbf{x}, \mathbf{z}) = \arg\min_{\{\mathbf{z}\}} g(\mathbf{z}) + \frac{1}{2\rho^2}\|\mathbf{x} - \mathbf{z}\|_2^2$$

- **Plug-and-Play** (PnP) $\Longleftarrow$ uses pretrained DL denoisers

# The augmented Bayesian approach

Asymptotically exact data augmentation :

$$p(\mathbf{x} \mid \mathbf{y}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{x})\right]$$
$$\Downarrow$$
$$p(\mathbf{x},\mathbf{z} \mid \mathbf{y}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{z})\right] \text{ knowing that } \mathbf{x} = \mathbf{z}$$

## Proposed approach

systematic & simple way to relax $p(\mathbf{x},\mathbf{z} \mid \mathbf{y})$ leading to efficient algorithms

$$\int p_\rho(\mathbf{x},\mathbf{z} \mid \mathbf{y})\mathrm{d}\mathbf{z} \underset{\rho\to 0}{\longrightarrow} p(\mathbf{x} \mid \mathbf{y})$$

$\longrightarrow$ How to build $p_\rho(\mathbf{x},\mathbf{z} \mid \mathbf{y})$ ?

# On building $p_\rho$

A natural manner to construct $p_\rho$ is to define:

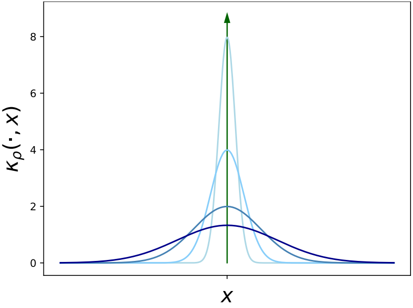
$$p_\rho(\mathbf{x}, \mathbf{z} \mid \mathbf{y}) \propto \exp\left[-f(\mathbf{x}, \mathbf{y}) - g(\mathbf{z})\right] \kappa_\rho(\mathbf{z}, \mathbf{x})$$

Where :

$$\kappa_\rho(\cdot, \mathbf{x}) \xrightarrow[\rho \to 0]{\text{weak}} \delta_{\mathbf{x}}(\cdot)$$

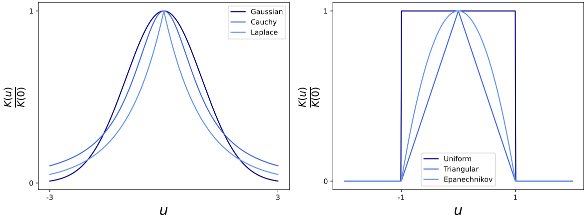
such that :

$$\|p - p_\rho\|_{\mathrm{TV}} \underset{\rho \to 0}{\to} 0$$



# Kernel choice $\kappa_\rho(\mathbf{z}, \boldsymbol{\theta}) \propto \rho^{-d} K\left(\frac{\mathbf{z}-\boldsymbol{\theta}}{\rho}\right)$

| name | support | $K(u)$ |
| --- | --- | --- |
| Gaussian | $\mathbb{R}$ | $\frac{1}{\sqrt{2\pi}} \exp\left(-u^2/2\right)$ |
| Cauchy | $\mathbb{R}$ | $\frac{1}{\pi(1+u^2)}$ |
| Laplace | $\mathbb{R}$ | $\frac{1}{2}\exp(-\lvert u\rvert)$ |
| Dirichlet | $\mathbb{R}$ | $\frac{\sin^2(u)}{\pi u^2}$ |
| Uniform | $[-1, 1]$ | $\frac{1}{2}\mathbb{I}_{\lvert u\rvert\leq 1}$ |
| Triangular | $[-1, 1]$ | $(1-\lvert u\rvert)\mathbb{I}_{\lvert u\rvert\leq 1}$ |
| Epanechnikov | $[-1, 1]$ | $\frac{3}{4}\left(1-u^2\right)\mathbb{I}_{\lvert u\rvert\leq 1}$ |




$\longrightarrow$ Similarity with "noisy" ABC methods (Marin et al. 2012; Wilkinson 2013)

# Bias associated with Gaussian smoothing

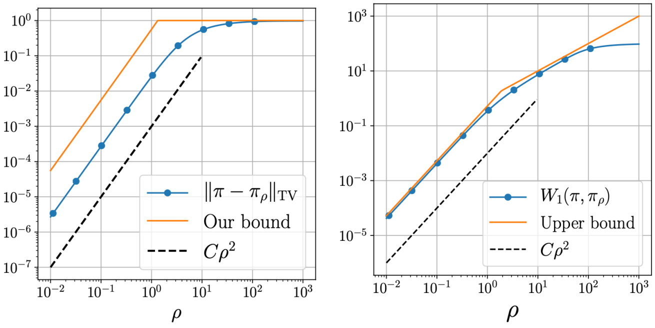
$$p_\rho(\mathbf{x} \mid \mathbf{y}) \propto \int_{\mathbb{R}^d} \exp\left[\underbrace{-f(\mathbf{x}, \mathbf{y}) - g(\mathbf{z})}_{U(x)} - \underbrace{\frac{1}{2\rho^2}\|\mathbf{x} - \mathbf{z}\|_2^2}_{\log \kappa_\rho}\right] d\mathbf{z}$$

| Distance | Upper bound | Main assumptions on $U(\cdot)$ |
|---|---|---|
| | $\rho 2\sqrt{d}L + o(\rho)$ | $L$-Lipschitz |
| $\|p_\rho - p\|_{\mathrm{TV}}$ | $\frac{1}{2}\rho^2 Md$ | $M$-smooth |
| | $\frac{1}{2}\rho^2 Md + o\left(\rho^2\right)$ | $M$-smooth & strongly convex |
| $W_1\left(p_\rho, p\right)$ | $\min\left(\rho\sqrt{d}, \frac{1}{2}\rho^2\sqrt{Md}\right)$ | $M$-smooth & strongly convex |

# Bias associated with Gaussian smoothing

Illustration of these bounds for an univariate Gaussian toy example :

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\mathbf{x}^2}{2\sigma^2}} \quad p_\rho(\mathbf{x}) = \frac{1}{\sqrt{2\pi\left(\sigma^2 + \rho^2\right)}} e^{-\frac{\mathbf{x}^2}{2(\sigma^2+\rho^2)}}$$



# Splitted Gibbs sampling: conditional distributions

**Goal sample from posterior**:

$$p(\mathbf{x}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{x})\right]$$
$$\Downarrow \text{Approx}$$
$$p_\rho(\mathbf{x},\mathbf{z}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{z}) - \frac{1}{2\rho^2}\|\mathbf{x}-\mathbf{z}\|_2^2\right]$$

Full conditional distributions under the split distribution $p_\rho$ :

$$p\left(\mathbf{x} \mid \mathbf{z},\mathbf{y};\rho^2\right) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - \frac{1}{2\rho^2}\|\mathbf{x}-\mathbf{z}\|_2^2\right] \quad (1)$$
$$p\left(\mathbf{z} \mid \mathbf{x};\rho^2\right) \propto \exp\left[-g(\mathbf{z}) - \frac{1}{2\rho^2}\|\mathbf{z}-\mathbf{x}\|_2^2\right] \quad (2)$$

**Note that** $f$ and $g$ are now separated in **2 distinct distributions**[2].

[2] Gibbs sampling: *Robert and Cassela, Springer 1999*

# Splitted Gibbs sampling: conditional distributions

**First Gibbs kernel** $\to$ task dependent $f(\mathbf{x}, \mathbf{y})$:

$$p\left(\mathbf{x} \mid \mathbf{z}, \mathbf{y}; \rho^2\right) \propto \exp\left[-f(\mathbf{x}, \mathbf{y}) - \frac{1}{2\rho^2} \|\mathbf{x} - \mathbf{z}\|_2^2\right]$$

Can be sampled using state of the art methods:

- Gaussian forward model: Fourier or Aux-V1 or E-PO[3]
- Other forward model: P-MYULA[4]= proximal MCMC[5]

[3] Hight dim Gaussian sampling: *Vono et al., SIAM Review, 2022*
[4] P-MYULA: *Durmus et al., SIAM IS, 2018*
[5] Proximal MCMC: *M. Pereyra, Stat Comput, 2016*

# Splitted Gibbs sampling: conditional distributions

**Second Gibbs kernel** $\rightarrow$ prior dependent $g(\mathbf{z})$:

$$p\left(\mathbf{z} \mid \mathbf{x}; \rho^2\right) \propto \exp\left[-g(\mathbf{z}) - \frac{1}{2\rho^2}\|\mathbf{z}-\mathbf{x}\|_2^2\right]$$

Posterior of a Gaussian corruption of $\mathbf{x}$ with prior $g(\mathbf{z})$

- Generative models: DVAE, DGAN, DDPM[6]

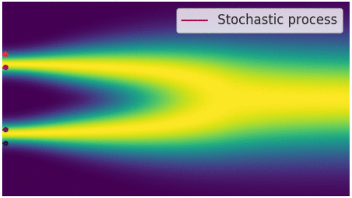
## Proposed approach

Replace ineficient explicit prior by a deep generative denoiser $\rightarrow$ DDPM.

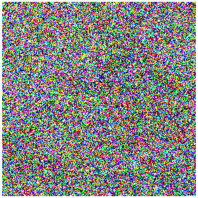
[6]DDPM: *Ho et al., NeurIPS, 2020*

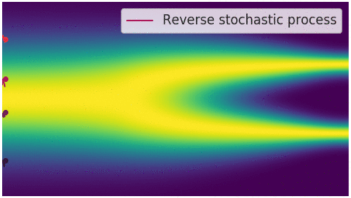
# Denoising Diffusion Probabilistic Models[7]

- Forward diffusion: $p\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(x_t; \sqrt{1-\beta(t)}x_{t-1}, \beta(t)\mathbf{I}\right)$




- Reverse denoising: $q_{\boldsymbol{\theta}}\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right)$






[7] Visual: *Song and Ermon, NeurIPS, 2019*

# DDPM as stochastic denoisers

The denoising step of SGS is done as follows:

1. Infer the std of the noisy input using an estimator $\hat{\sigma} = \Phi(x)$
2. Deduce the starting time $\widehat{t^*} = \alpha^{-1}(\widehat{\sigma}^2)$
3. Start the backward diffusion at $\widehat{t^*}$ and denoise using $q_{\boldsymbol{\theta}}\left(x_{t-1} \mid x_t\right)$

Note that

- a noise level $\alpha(t^*)$ is associated to a unique instant $t^*$
- $t^*$ adjusts the amount of imposed regularization

# DDPM as stochastic denoisers

Transition from the noise-free image $x_0$ to noisy image $x_t$:

$$p\left(x_t \mid x_0\right) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}(t)}x_0, \alpha(t)\mathbf{I}\right)$$

$$\text{where} \quad \alpha(t) = \prod_{j=1}^{t}(1-\beta(j)) \quad \text{and} \quad \bar{\alpha}(t) = 1-\alpha(t)$$

Any image $x_t$ is a noisy version of $x_0$ corrupted by a Gaussian noise of covariance $\alpha(t)\mathbf{I}$.

# Algorithm

**Algorithm 1:** PnP-SGS using DDPM

**Input :** Parameter $\rho^2$, total number of iterations $N_{\mathrm{MC}}$, number of burn-in iterations $N_{\mathrm{bi}}$, pre-trainted DDPM $s_{\boldsymbol{\theta}}(\cdot,\cdot)$, scheduling variance function $\alpha(\cdot)$, initialization $\mathbf{z}^{(0)}$

**for** $n \leftarrow 1$ **to** $N_{\mathrm{MC}}$ **do**

  *# Sampling the variable of interest* $\mathbf{x}^{(n)}$

  Draw $\mathbf{x}^{(n)} \sim p(\mathbf{x} \mid \mathbf{z}, \mathbf{y}; \rho^2)$ according to (6)

  *# Estimating noise level in* $\mathbf{x}^{(n)}$

  Set $\hat{\sigma} = \Phi(\mathbf{x}^{(n)})$ using [38]

  *# Setting the number of diffusion steps to denoise* $\mathbf{x}^{(n)}$

  Set $\widehat{t^*} = \alpha^{-1}(\hat{\sigma}^2)$

  *# Sampling the splitting variable* $\mathbf{z}^{(n)}$ *according to (7)*

  Set $\mathbf{u}_{\widehat{t^*}} = \mathbf{x}^{(n)}$

  **for** $j \leftarrow \widehat{t^*}$ **downto** 1 **do**

    Draw $\mathbf{u}_{j-1} \sim q_{\boldsymbol{\theta}}(\mathbf{u}_{j-1} \mid \mathbf{u}_j)$ according to (10)

  **end for**

  Set $\mathbf{z}^{(n)} = \mathbf{u}_0$

**end for**

**Output:** Collection of samples $\left\{\mathbf{x}^{(n)}, \mathbf{z}^{(n)}\right\}_{t=N_{\mathrm{bi}+1}}^{N_{\mathrm{MC}}}$ asymptotically distributed according to (4).

# Mixing time

How many SGS iterations should I run to obtain a sample from $p(\mathbf{x} \mid \mathbf{y})$?

This question can be answered by giving an upper bound on the so-called $\epsilon$-mixing time:

$$t_{\mathrm{mix}}(\epsilon; \nu) = \min \left\{ t \geq 0 \mid D\left(\nu P_{\mathrm{SGS}}^{t}, p\right) \leq \epsilon \right\}$$

where $\epsilon > 0, \nu$ is an initial distibution, $P_{\mathrm{SGS}}$ is the Markov kernel of SGS and $D$ is a statistical distance.

$\longrightarrow$ We are here interested in providing explicit bounds w.r.t. $d, \epsilon$ and regularity constants associated to $U$[8].

[8] *Vono et al., JMLR*

# Explicit convergence rates

## Assumptions

$U$ is $M$-smooth & $m$-strongly convex, starting from $\nu = \mathcal{N}\left(\mathbf{x}^\star, M^{-1}\mathbf{I}_d\right)$

- $\forall \epsilon \in (0,1)$, with $\rho^2 \leq \frac{\epsilon}{dM}$
- for $t \geq t_{\mathrm{mix}}(\epsilon; \nu)$ where :

$$t_{\mathrm{mix}}(\epsilon;\nu) = \frac{\log\left(\frac{2}{\epsilon}\right) + C/2}{K_{\mathrm{SGS}}}$$

Where :

$$K_{\mathrm{SGS}} = \frac{m\rho^2}{1+m\rho^2} \quad \text{and} \quad C = \frac{5d}{8} + \frac{d}{2}\log\left(\frac{M}{m}\right)$$

- We have :

$$\left\|\nu P_{\mathrm{SGS}}^t - p\right\|_{\mathrm{TV}} \leq \epsilon$$

# Image Inpainting

We consider the following linear model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

the likelihood function associated with the observation $\mathbf{y}$ writes

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left[-\frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{y})^T \boldsymbol{\Omega}(\mathbf{H}\mathbf{x} - \mathbf{y})\right].$$

Where $\mathbf{H} \in \{0,1\}^{N\times M}$is a binary matrix associated and $\boldsymbol{\Omega}^{-1} = \sigma^2 \mathbf{I}_M$.

# Image Inpainting

the proposed PnP-SGS algorithm yields the conditional distribution defined here as

$$p(\mathbf{x} \mid \mathbf{z}, \mathbf{y}; \rho^2) = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \mathbf{Q}_{\mathbf{x}}^{-1}\right)$$

with

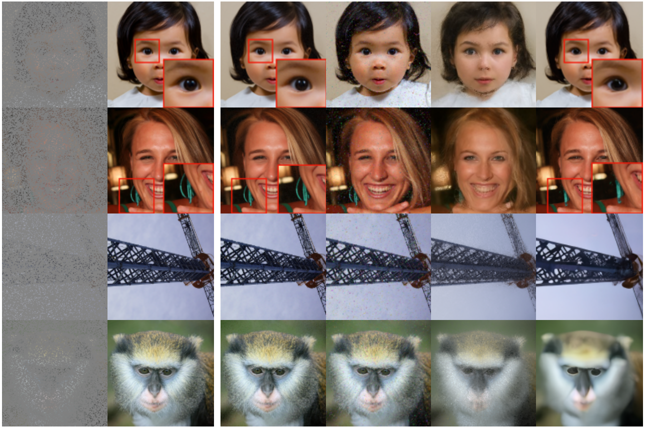
$$\begin{cases} \mathbf{Q}_{\mathbf{x}} = \frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H} + \frac{1}{\rho^2}\mathbf{I}_N \\ \boldsymbol{\mu}_{\mathbf{x}} = \mathbf{Q}_{\mathbf{x}}^{-1}\left(\frac{1}{\sigma^2}\mathbf{H}^T\mathbf{y} + \frac{1}{\rho^2}\mathbf{z}\right). \end{cases}$$

From the Sherman-Morrison-Woodbury formula, we have:

$$\mathbf{Q}_{\mathbf{x}}^{-1} = \rho^2\left(\mathbf{I}_N - \frac{\rho^2}{\sigma^2+\rho^2}\mathbf{H}^T\mathbf{H}\right).$$

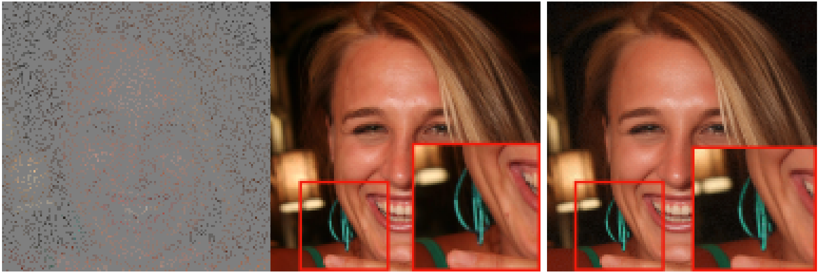
Which can be conducted efficiently with the exact perturbation-optimization (E-PO) algorithm.

# Image inpainting



From left to right: measure, true, PnP-SGS, TV-SGS, DDRM, MCG.

# Image inpainting



From left to right: measure, true, PnP-SGS (MMSE)

# MMSE estimation

From top to bottom: Gaussian blur, motion blur, superresolution.

# MMSE estimation

# Quantitative evaluation

| | FID ↓ | LPIPS ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| PnP-SGS | **38.36** | **0.144** | **23.59** | **0.813** |
| TV-SGS | 71.12 | 0.785 | 21.09 | 0.524 |
| PnP-ADMM | 123.61 | 0.692 | <u>22.41</u> | 0.325 |
| TV-ADMM | 181.56 | 0.463 | 22.03 | <u>0.784</u> |
| Score-SDE | 76.54 | 0.612 | 13.52 | 0.437 |
| DDRM | 69.71 | 0.587 | 9.19 | 0.319 |
| MCG | <u>39.26</u> | <u>0.286</u> | 21.57 | 0.751 |

Benchmark for FFHQ $256 \times 256$ with 1000 images.

| PnP-SGS | TV-SGS | PnP-ADMM | Score-SDE | DDRM | MCG |
|---|---|---|---|---|---|
| 13.81 | 218.90 | 3.63 | 36.71 | 29.03 | 80.10 |

Inpainting: computational times (s.)

# Black Holes image reconstruction - VLBI interferometer

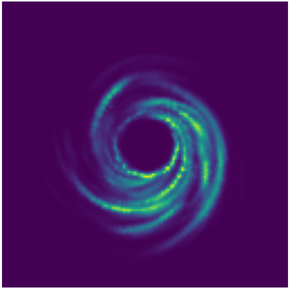
The Event Horizon Telescope measures complex-visibility $V$ :

$$V(u,v) = \int I(x,y) e^{-2\pi i(ux+vy)} dxdy$$

Signal corrupted by the atmosphere, with the complex term $g$ :

$$\tilde{V}_{ij} = g_i g_j^* V_{ij}$$

and Complex Gaussian thermal noise.



# Conclusion: PnP-SGS

- Plug-and-play split Gibbs sampler (PnP-SGS):
  - Bridge MCMC and Plug-and-Play methods
  - Explicit convergence rates and mixing times
  - Credibility intervals
  - Fast and scalable sampling
  - general: using any deep generative denoiser
  - State of the art: on image processing problems
- Perspectives:
  - Study theoretical denoising properties of deep generative models
  - Adaptive PnP-SGS : find a sequence $\rho_{k\in\mathcal{N}}$

# Thank you for your attention!
# Questions ?




Joint work with Nicolas Dobigeon, Pierre Chainais, Arnaud Doucet & Paul Tiede

# Appendix

# Poisson and non-Gaussian

Let consider the observation of some image $\mathbf{y} \in \mathbb{N}^m$,damaged and contaminated by Poisson noise

$$y_i \overset{\text{iid.}}{\sim} \mathcal{P}\left([\mathbf{Hx} + \mathbf{b}]_i\right)$$

the likelihood distribution is rewritten as follows

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left(-f_1(\mathbf{Hx}; \mathbf{y})\right)$$

where

$$f_1(\mathbf{Hx}; \mathbf{y}) = \sum_{i=1}^{m} -y_i \log\left([\mathbf{Hx} + \mathbf{b}]_i\right) + [\mathbf{Hx} + \mathbf{b}]_i$$

By the application of Bayes' rule, the AXDA envelope of the posterior distribution writes

$$\pi_\rho(\mathbf{x}, \mathbf{z}) \propto \exp\left(-f_1(\mathbf{Hx}) - g(\mathbf{z}) + \frac{1}{2\rho}^2 \|\mathbf{x} - \mathbf{z}\|_2^2\right)$$

## Comparison with ULA

| Distance | Upper Bound | Main assumptions |
|---|---|---|
| $\lVert\pi_\rho - \pi\rVert_{\mathrm{tv}}$ | $\frac{1}{2}\rho^2 M_1 d$ | $M-$smooth |
| $\lVert\pi_{\mathrm{ULA}} - \pi\rVert_{\mathrm{tv}}$ | $\sqrt{2Md}\cdot\rho$ | $M$-smooth & convex |
| $W_1(\pi_\rho, \pi)$ | $\min\left(\rho\sqrt{d}, \frac{1}{2}\rho^2\sqrt{Md}\right)$ | $M$-smooth, $m$-strongly convex |
| $W_1(\pi_{\mathrm{ULA}}, \pi)$ | $\sqrt{2\frac{M}{m}d}\cdot\rho$ | $M$-smooth, $m$-strongly convex |

$\longrightarrow$ Higher order approximation based on the explicit form of $\pi_\rho$.

# Evolution of $\hat{t}$

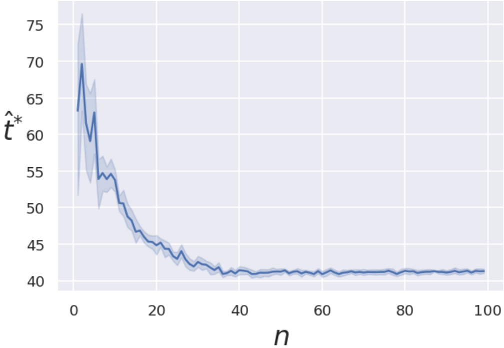
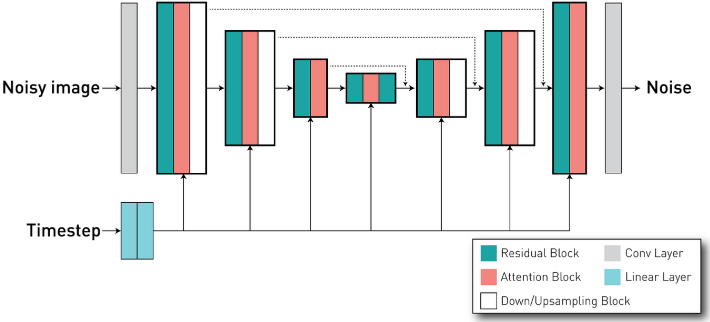



Figure: Inpainting: evolution of $\widehat{t^*}$ along the PnP-SGS iterations ($T = 1000$). Results have been averaged over 100 runs conducted on the same image. Shaded areas stand for the corresponding standard deviation.

# Diffusion model implementation considerations

Diffusion models often use U-Net architectures with ResNet blocks and self-attention




Time representation: sinusoidal positional embeddings or random Fourier features.