Monte Carlo sampling and deep generative models for Bayesian inference

Florentin Coeurdoux

November 23th, 2023

Bayesian inference

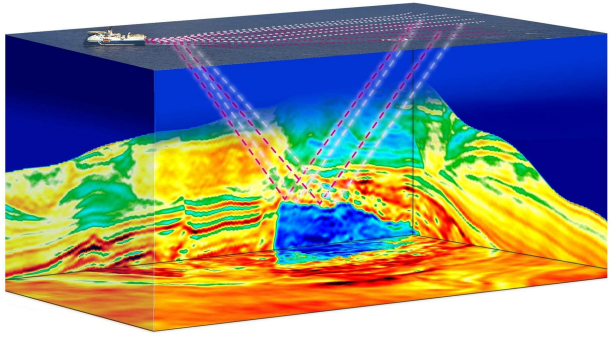
$\mathbf{x}$: unknown object of interest
$\mathbf{y}$: available data (observation)

Update of prior information with available data[1]:

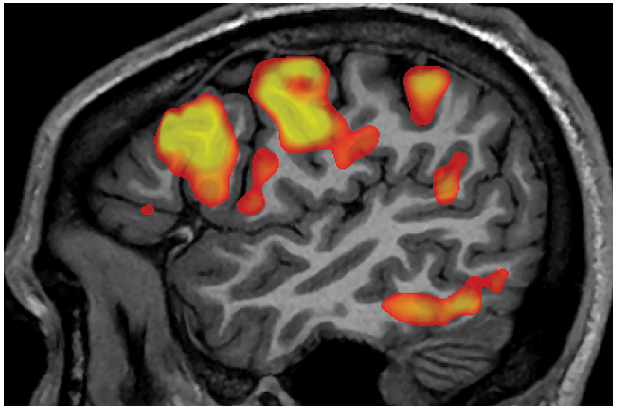
Prior		Likelihood		Posterior
$p(\mathbf{x})$	$\times$	$p(\mathbf{y} \mid \mathbf{x})$	$\rightarrow$	$p(\mathbf{x} \mid \mathbf{y})$
$\uparrow$		$\uparrow$		$\uparrow$
pre-observation knowledge		acquisition process		best guess

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

Various applications

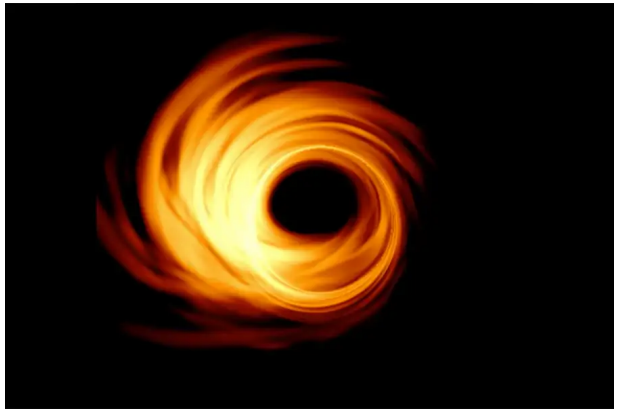


(a) Seismic imaging



(b) Brain fMRI

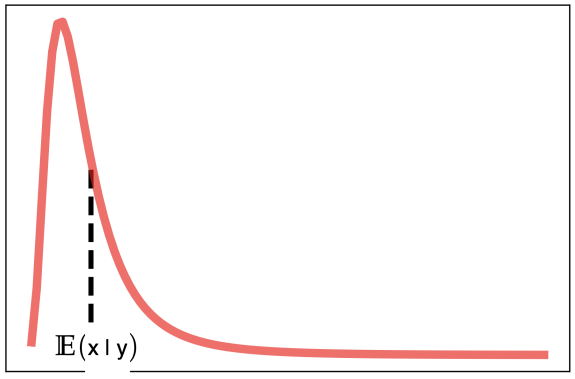
(c) Earth observation



(d) Astrophysics

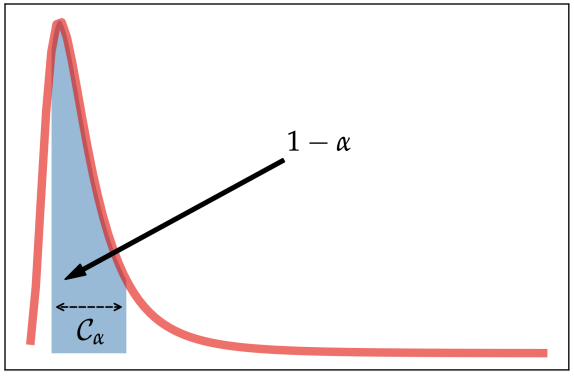
Bayesian inference : Uncertainty quantification

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

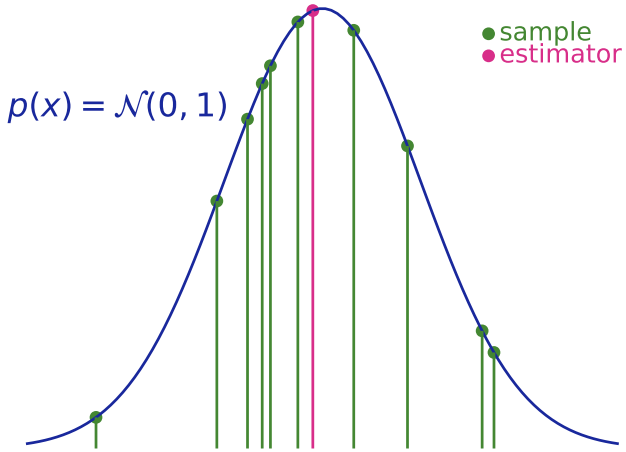
Monte Carlo integration

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

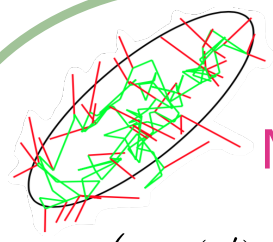
Starting hypothesis

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

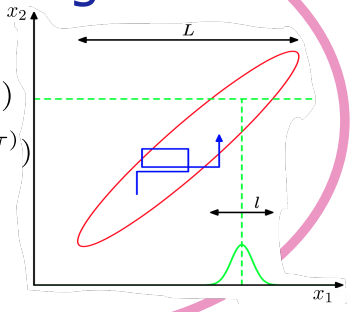


Metropolis-Hastings

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

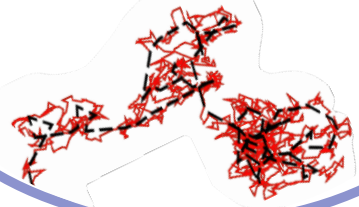
Gibbs sampling

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



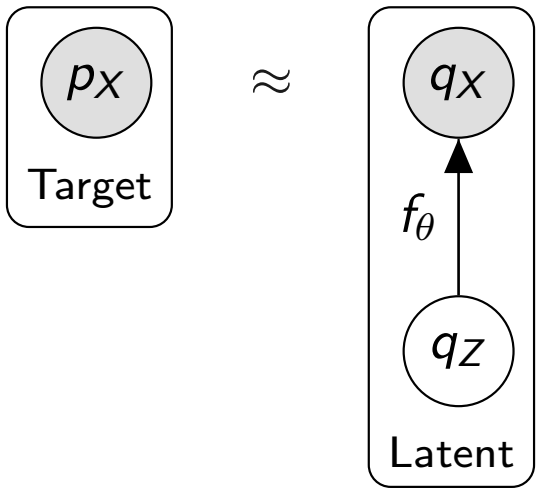
Langevin sampling

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

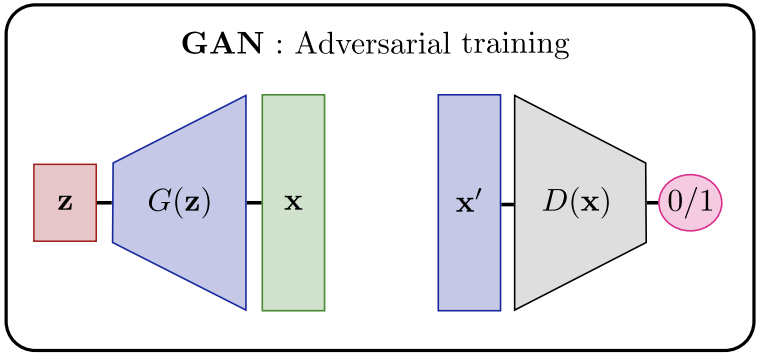


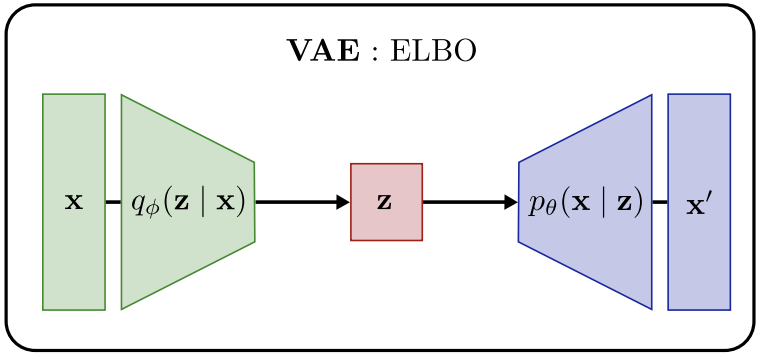
What is deep generative modeling ?

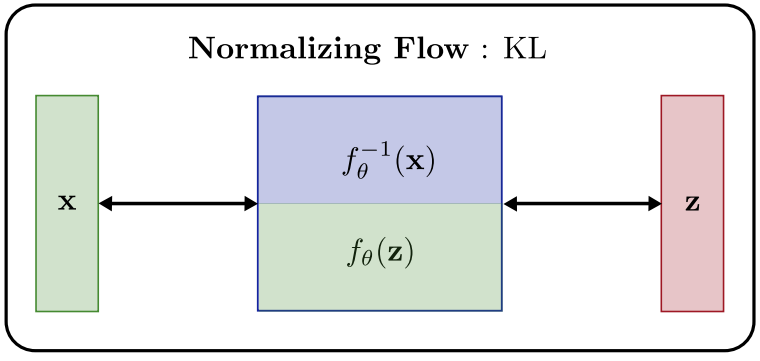
- Aim: Approximate distribution $q_X \approx p_X$ such that $q_X := f_{\theta\sharp} q_Z$
- A general approach :
  - Samples $\{x_i\}_{i=0}^N$ from p_X defined over $\mathcal{X}$.
  - Choose a simple latent distribution q_Z defined over $\mathcal{Z}$.
  - Learn a mapping $f_\theta : \mathcal{Z} \to \mathcal{X}$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

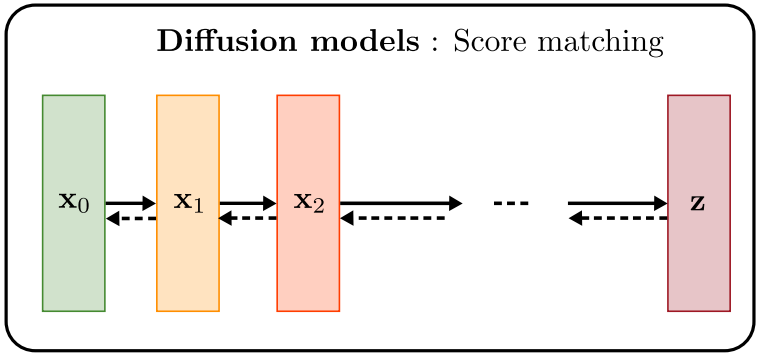


A myriad of models





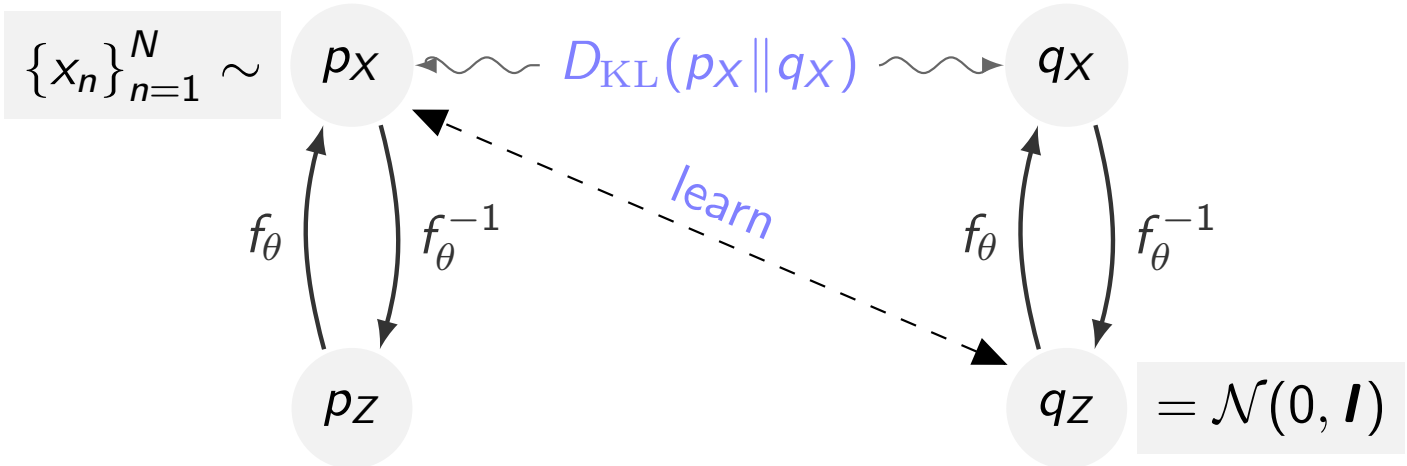




Normalizing flows

Architecture

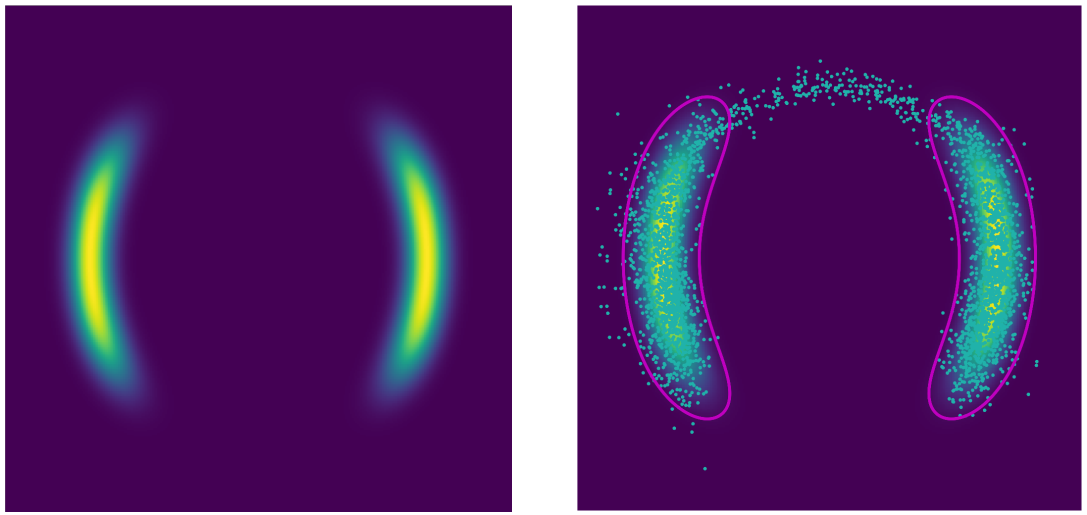
- Bijective: $f_\theta(\mathbf{z}) = \mathbf{x}$ and $f_\theta^{-1}(\mathbf{x}) = \mathbf{z}$
- Aim: $p_X \approx q_X := f_\sharp q_Z$
- Types: RealNVP, MAF, GLOW, NICE, FFJORD, $\cdots$
- Change of variables formula: $q_X(\mathbf{x}) = q_Z\left(f_\theta^{-1}(\mathbf{x})\right)\left|J_{f_\theta^{-1}}(\mathbf{x})\right|$



Problem statement

Pathological behaviors of flow based models:

- Incapacity to model multi-components distributions
- Prescribes high likelihood to OOD data.



p_X q_X

Topological mismatch

- The mapping f is continuous, bijective and differentiable.
- q_X and q_Z are homeomorphic $\rightarrow$ maintain topological properties.
- Then the push-forward operator $f_\sharp$ can yield $q_X = p_X$ only if

$$\operatorname{supp}(p_X) = \overline{f(\operatorname{supp}(q_Z))}$$

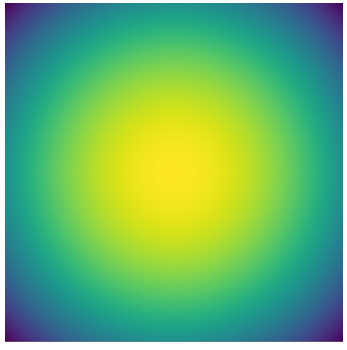
where $\bar{B}$ is the closure of set B.

Proposition

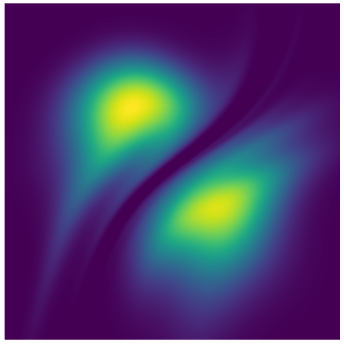
Let q_Z and p_X denote distributions defined on $\mathbb{R}^d$. Assume that $\operatorname{supp}(q_Z) \neq \operatorname{supp}(p_X)$. For any sequence of measurable, differentiable Lipschitz functions $f_t : \mathbb{R}^{d_Z} \rightarrow \mathbb{R}^{d_X}$, if $f_{t\sharp} q_Z \xrightarrow{\mathcal{D}} p_X$ when $t \rightarrow +\infty$, then

$$\lim_{t \rightarrow \infty} \sup_{z \in \mathcal{Z}} (\| J_{f_t}(z) \|) = +\infty$$

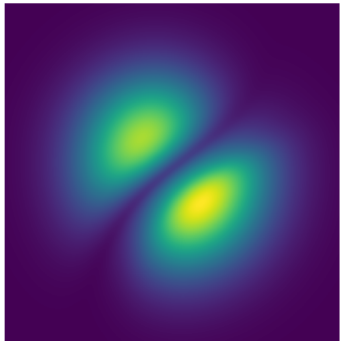
Topological mismatch



$q_Z = \mathcal{N}(0, I)$



$p_Z := f_\sharp^{-1} p_X$



$\tilde{q}_Z(z) = q_Z(z)\,|J_f(z)|^{-1}$

Implications :

- NF partitions the latent space in regions separated by exploding J_f.
- Zones with exploding J_f correspond to in between modes with low probability areas in p_X.

$\rightarrow$ Sample by avoiding high J_f areas

Global kernel

NF models push any distribution toward $\mathcal{N}(0, I)$, making it a perfect proposal distribution in a Metropolis-Hastings. The corresponding acceptance ratio writes:

$$\begin{aligned}\alpha_{\mathrm{I-MH}}\left(\mathbf{z}_k, \mathbf{z}'\right) &= \min\left(1, \frac{\tilde{q}_Z\left(\mathbf{z}'\right) q_Z\left(\mathbf{z}_k\right)}{\tilde{q}_Z\left(\mathbf{z}_k\right) q_Z\left(\mathbf{z}'\right)}\right) \\ &= \min\left(1, \frac{\left|J_f\left(\mathbf{z}_k\right)\right|}{\left|J_f\left(\mathbf{z}'\right)\right|}\right)\end{aligned}$$

Algorithm 4: Sampling kernel $\mathcal{K}_{\mathrm{I-MH}}(\cdot)$.

Input: trained NF $f(\cdot)$, current state z_k of the sampler.

`/* Draw candidate */`

Draw $z' \sim \mathcal{N}(0, 1)$

`/* Accept/reject procedure */`

Draw $u \sim \mathcal{U}(0, 1)$

if $u < \alpha_{\mathrm{I-MH}}(z_k, z')$ **then**

  Set $z_{k+1} = z'$

else

  Set $z_{k+1} = z_k$

Output: New state $z_{k+1} = \mathcal{K}_{\mathrm{I-MH}}(z_k)$ of the sampler.

Sampling in the latent space

The Unadjusted Langevin Algorithm (ULA) which targets q_X:

$$dX_t = \nabla_x \log q_X(X_t)dt + \sigma_t dB_t$$

Following Îto's lemma with second order truncation :

$$df^{-1}(X_t) = J_{f^{-1}}(X_t)\nabla_x \log q_X(X_t)dt + \sigma_t J_{f^{-1}}(X_t)dB_t$$

which can be rewriten as :

$$\begin{aligned} dZ_t &= J_f^{-1}(Z_t)J_f^{-1}(Z_t)\nabla_z \log \tilde{q}_Z(Z_t)dt + \sigma_t J_f^{-1}(Z_t)J_f^{-1}(Z_t)dB_t \\ &= \boxed{G^{-1}(Z_t)\nabla_z \log \tilde{q}_Z(Z_t)dt + \sigma_t \sqrt{G^{-1}(Z_t)}dB_t} \end{aligned}$$

With $G^{-1}(z) = \left[J_f^{-1}(z)\right]^2$ and $\tilde{q}_Z(z) = q_Z(z)\left|J_f(z)\right|^{-1}$.

$\rightarrow$ We recover the Riemann Manifold ULA which targets $\tilde{q}_Z$.

Riemannian manifold ajusted Langevin algorithm

- Discretization: Euler–Maruyama :

$$\mathbf{z}' = \mathbf{z} + \frac{\epsilon^2}{2} \cdot G^{-1}(\mathbf{z}) \nabla_{\mathbf{z}} \log \tilde{q}_Z(\mathbf{z}) + \epsilon \cdot \sqrt{G^{-1}(\mathbf{z})}\xi, \quad \xi \sim \mathcal{N}(0, I)$$

- Correction: Metropolis-Hasting rejection step :

$$g\left(\mathbf{z}' \mid \mathbf{z}\right) \propto \left|J_f(\mathbf{z})\right| \exp\left[-\frac{1}{2\epsilon^2}\left\| J_f(\mathbf{z})\left(\mathbf{z}' - \mathbf{z}\right) + \frac{\epsilon^2}{2} J_f^{-1}(\mathbf{z}) \nabla_{\mathbf{z}} \log \tilde{q}_Z(\mathbf{z})\right\|^2\right]$$

The samples are accepted with probability

$$\alpha_{\mathrm{RMMALA}}\left(\mathbf{z}_k, \mathbf{z}'\right) = \min\left(1, \frac{\tilde{q}_Z\left(\mathbf{z}'\right) g\left(\mathbf{z}_k \mid \mathbf{z}'\right)}{\tilde{q}_Z\left(\mathbf{z}_k\right) g\left(\mathbf{z}' \mid \mathbf{z}_k\right)}\right).$$

Efficient implementation

- **Efficient diffusion:**
  - Proposal scheme requires sampling from $\mathcal{N}(0, \epsilon^2 G^{-1}(\cdot))$ → slow
  - Alternatively, we take advantage of the 1st order expansion:

$$f^{-1}\left(\underbrace{f\left(\mathbf{z}_k\right)+\epsilon\xi}_{\mathbf{x}_k+\epsilon\xi}\right) \simeq \underbrace{\mathbf{z}_k}_{f^{-1}\circ f(\mathbf{z}_k)} + \epsilon J_{f^{-1}}\left(\underbrace{f\left(\mathbf{z}_k\right)}_{\mathbf{x}_k}\right)\xi$$

  → much faster to leverage the learned mapping f

- **Efficient latent score computation**
  Latent score $\tilde{s}_Z(\mathbf{z}) = J_f^{-1}(\mathbf{z})\nabla_z \log \tilde{q}_Z(z)$ appears in proposal and diffusion.
  - Jacobian computation $J_f^{-1}(\mathbf{z})$ is slow in high dimension.
  - The latent score can be expressed in the latent domain:

$$\nabla_{\mathbf{x}} \log q_X(\mathbf{x}) = J_f^{-1}(\mathbf{z}) \cdot \nabla_{\mathbf{z}} \log \tilde{q}_Z(\mathbf{z}) \quad \text{s.t} \quad \mathbf{x} = f(\mathbf{z})$$

Local kernel $\rightarrow$ RMMALA

Putting all together we have the following RMMALA:

Algorithm 3: Sampling kernel $\mathcal{K}_{\text{RMMALA}}(\cdot)$.

Input: trained NF $f(\cdot)$, time step ϵ, current state z_k of the sampler.

`/* Draw the candidate */`

Draw $\xi \sim \mathcal{N}(0, 1)$

Set $z' = f^{-1}\left(f(z_k) + \epsilon \cdot \xi\right) + \frac{\epsilon^2}{2} J_f^{-1}(z_k)\tilde{s}_Z(z_k)$

`/* Accept/reject procedure */`

Draw $u \sim \mathcal{U}(0, 1)$

if $u < \alpha_{\text{RMMALA}}(z_k, z')$ **then**

  Set $z_{k+1} = z'$

else

  Set $z_{k+1} = z_k$

Output: New state $z_{k+1} = \mathcal{K}_{\text{RMMALA}}(z_k)$ of the sampler.

$\rightarrow$ Only sample local region, get stuck in each modes

NF-SAILS

NF-SAILS combines the transition kernels $\mathcal{K}_{\text{RMMALA}}$ and $\mathcal{K}_{\text{I-MH}}$:

- Local search with probability p.
- Global search with probability $(1 - p)$.

Algorithm 5: NF-SAILS: NF SAmpling In the Latent Space.

Input: trained NF $f(\cdot)$, time step ϵ, probability p

`/* Initialization */`

Draw $z_0 \sim \pi_0(z)$

for $k = 0$ **to** K **do**

  `/* Choose the kernel */`

  Draw $u \sim \mathcal{U}(0, 1)$

  if $u < p$ **then**

    `/* LOCAL EXPLORATION (see Algo. 3) */`

    $z_{k+1} = \mathcal{K}_{\text{RMMALA}}(z_k)$

  else

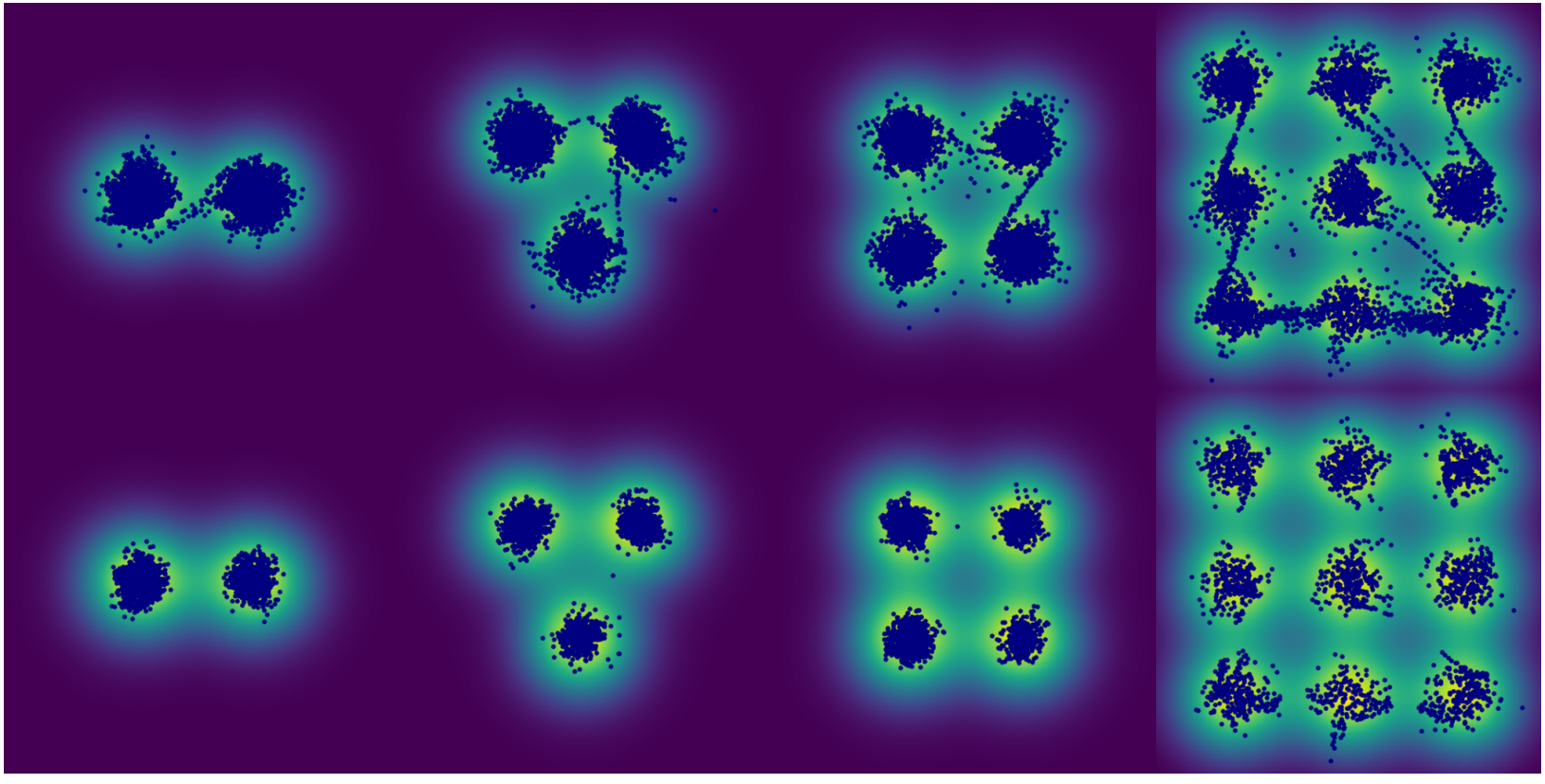
    `/* GLOBAL EXPLORATION (see Algo. 4) */`

    $z_{k+1} = \mathcal{K}_{\text{I-MH}}(z_k)$

end

Output: Collection of samples $\{z_k\}_{k=1}^{K}$.

Multimodal toy examples



(top) Ancestral Sampling (Bottom) NF-SAILS

High dimensional experiments

	↓ bpd	↓ FID
Naive sampling	3.35	44.6
NF-SAILS	**3.08**	43.11
WGAN-GP	N/A	**18.8**

Cifar10

	↓ bpd	↓ FID
Naive sampling	1.03	15.82
NF-SAILS	**0.96**	14.12
WGAN-GP	N/A	**12.89**

CelebA

	↓ bpd	↓ FID
Naive sampling	2.38	8.91
NF-SAILS	**2.11**	**7.86**
WGAN-GP	N/A	9.56

LSUN (bedroom)

Conclusion: NF-SAILS

Problem

- Jacobian norm explodes for disjoint support distributions
- Existence of pathological areas in the latent space

Solutions

- Langevin sampling procedure over the latent space
- Riemanian manifold MALA
- Applied to any pre-trained NF

Coeurdoux et al. *"Normalizing flow sampling with Langevin dynamics in the latent space"*, Submitted to Machine Learning

Workshop : Interfacing Bayesian statistics and machine learning, *"Langevin based Normalizing flow sampling"*, Bayes Centre, Edinburgh, Jan 2023.

Plug-and-Play split Gibbs sampler: Motivation

Inverse Problem

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

- **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 Bayesian approach augmented by splitting

- Asymp. eXact Data Augmentation (AXDA)[2]:

$$\pi(\mathbf{x}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{x})\right]$$

$$\Downarrow$$

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

$$\Downarrow$$

$$\pi_\rho(\mathbf{x},\mathbf{z}) \propto \exp\left[-f(\mathbf{x},\mathbf{y}) - g(\mathbf{z}) - \tfrac{1}{2\rho^2}\|\mathbf{x}-\mathbf{z}\|_2^2\right]$$

- Asymptotic convergence in distribution:

$$\|\pi - \pi_\rho\|_{\mathrm{TV}} \underset{\rho^2 \to 0}{\to} 0$$

[2]AXDA: *Vono et al, Journal of Computational and Graphical Statistics, 2020*

Splitted Gibbs sampling: conditional distributions

Full conditional distributions under the split distribution π_ρ :

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

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

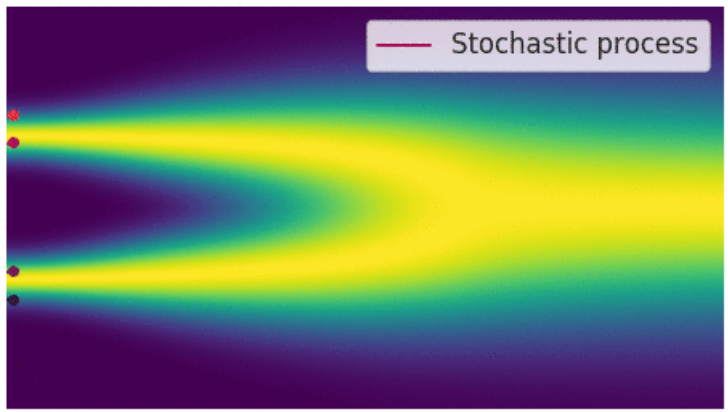
Note that f and g are now separated in **2 distinct distributions**[3].

- Eq (1) can be sampled using state of the art methods:
  - Gaussian variables: Fourier or Aux-V1 or E-PO
  - P-MYULA = proximal MCMC
- Eq (2) is the Posterior of a Gaussian corruption of $\mathbf{x}$ with prior $g(\mathbf{z})$
  - Generative models: DVAE, DGAN, DDPM

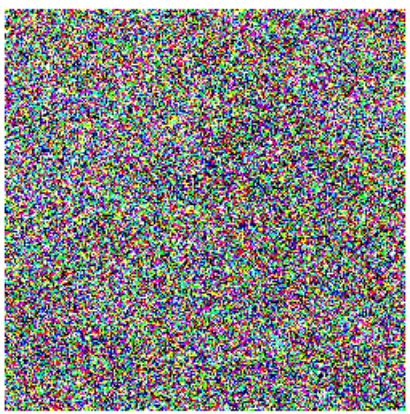
[3]SPA: *Vono et al, IEEE Transactions on Signal Processing, 2019*

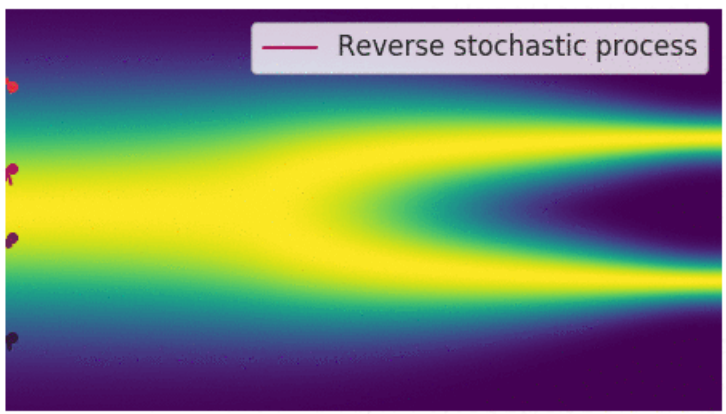
Denoising Diffusion Probabilistic Models[4]

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





[4]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 $q_{\theta}\left(x_{t-1} \mid x_t\right)$

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:

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

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

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

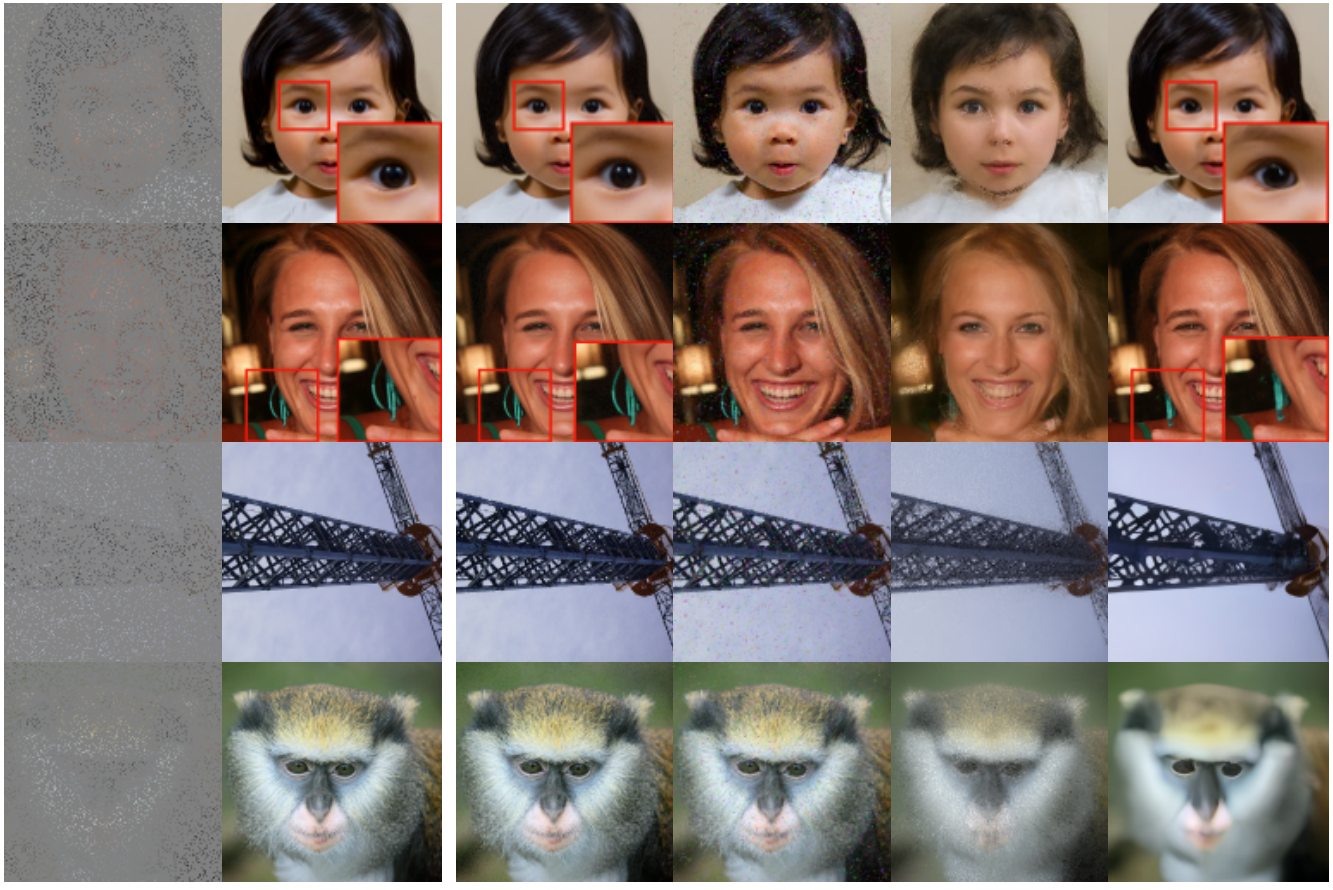
Input : Parameter ρ^2, total number of iterations N_{MC}, number of burn-in iterations N_{bi}, pre-trainted DDPM $s_{\boldsymbol{\theta}}(\cdot,\cdot)$, scheduling variance function $\alpha(\cdot)$, initialization $\mathbf{z}^{(0)}$

for $n \leftarrow 1$ **to** N_{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).

Image inpainting



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

MMSE estimation

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

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 256×256 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.)

Conclusion: PnP-SGS

The plug-and-play split Gibbs sampler (PnP-SGS) provides:

- Bridge AXDA and Plug-and-Play methods
- Asymptotic convergence toward the posterior
- Credibility intervals
- Fast and scalable sampling
- General: using any deep generative denoiser.

$\rightarrow$ No theoretical denoising properties of diffusion models yet

Contributions

Submitted journal articles

- Coeurdoux et al. *"Normalizing flow sampling with Langevin dynamics in the latent space"*, Submitted to Machine Learning
- Coeurdoux et al. *"Plug-and-Play split Gibbs sampler: embedding deep generative priors in Bayesian inference"*, Submitted to 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

...