Plug&Play sampling for inverse problems in imaging

Julie Delon (with Rémi Laumont, Andrés Almansa, Marcelo Pereyra, Valentin de Bortoli)

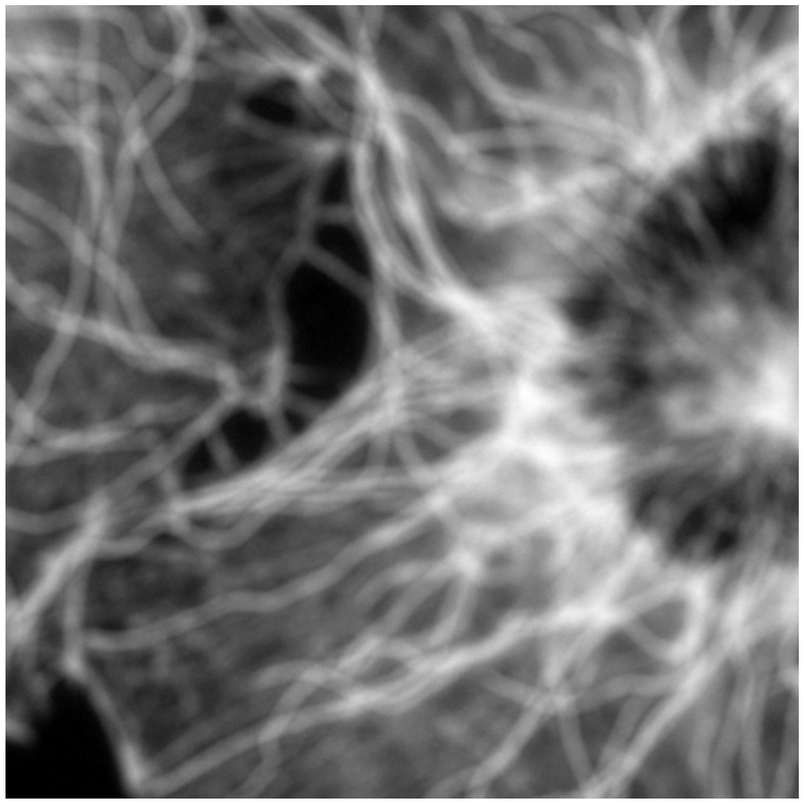
Mathematical Models for Plug-and-play Image Restoration

December, 7-8th, 2022

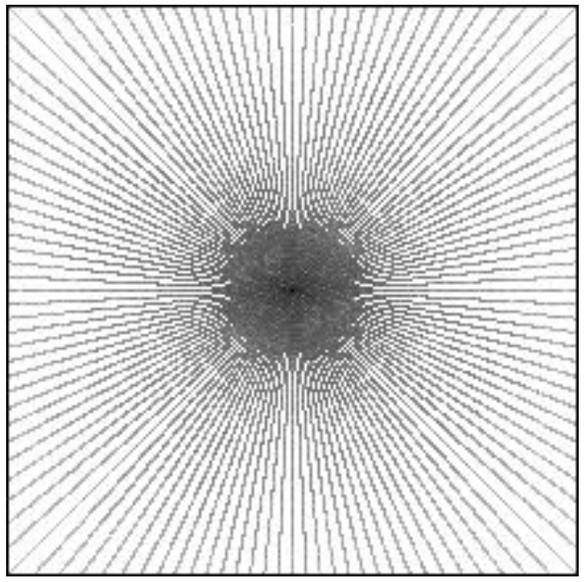
LABORATOIRE MAP5

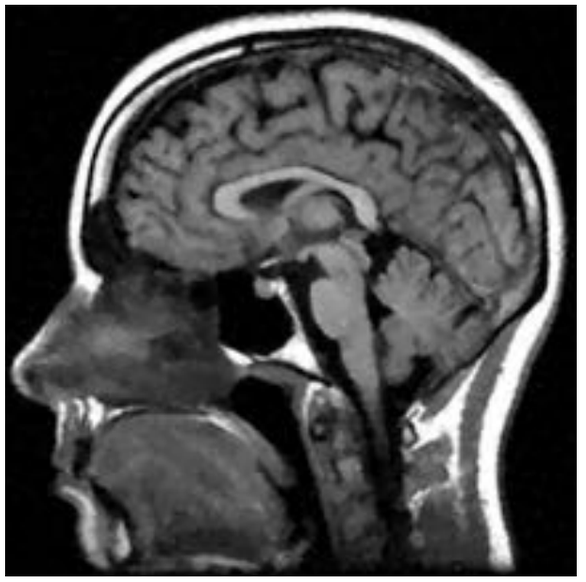
MIA

Inverse problems in imaging



R. Abergel,
L. Moisan





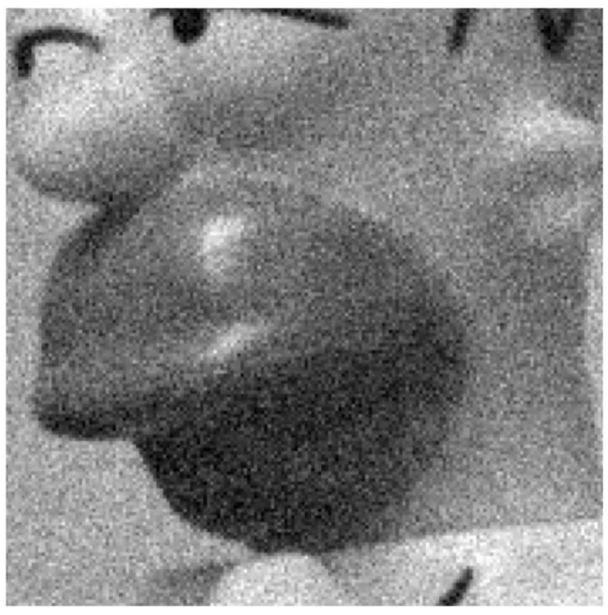
[The Mandalorian, Disney, 2020]

Inverse problems

$$\underset{\mathbb{R}^d}{y} = \underset{\mathbb{R}^{d\times n}}{A}\underset{\mathbb{R}^n}{x} + \underset{\mathbb{R}^d}{n}$$

observation — degradation operator — unknown — noise

image x

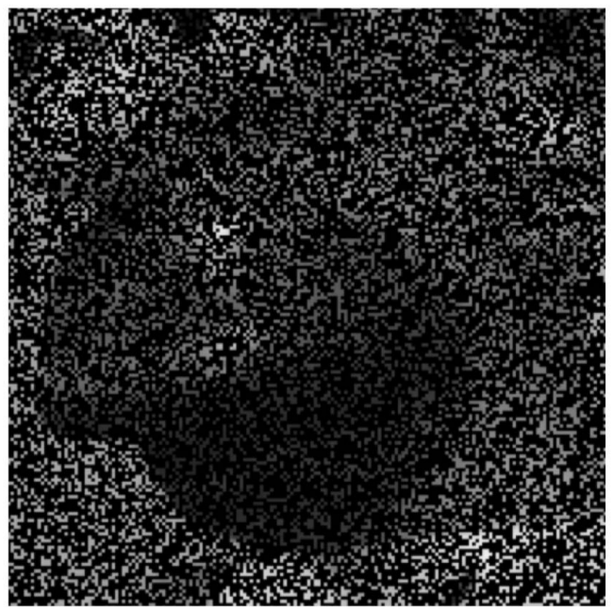


additive noise

$$A = I_n$$

blur

$$A = \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_{n-1} & a_0 & \dots & a_{n-2} \\ a_1 & a_2 & \dots & a_0 \end{pmatrix}$$



missing pixels

$$A = \begin{pmatrix} ? & 0 & \dots & 0 \\ 0 & ? & \dots & 0 \\ 0 & 0 & \dots & ? \end{pmatrix}$$

Goal : estimate x from the observation y

A poorly conditioned or rank deficient!

Bayesian framework

Forward degradation model

$$Y = AX + N$$

- x seen as a realization of a r.v. X with **prior distribution** $p_X(x) \propto e^{-U(x)}$
- y realization of $Y \,|\, X = x$, **likelihood** $p_{Y|X}(y \,|\, x) = p_N(y - Ax) \propto e^{-F(x,y)}$, with p_N the noise distribution
- Inferences on x based on the **posterior**

$$p_{X|Y} = \frac{p_{Y|X} p_X}{p_Y} \propto p_{Y|X} p_X$$

$\rightarrow$ models our belief about X after observing $Y = y$.

Bayesian framework

1. **MAP** estimation (optimisation)

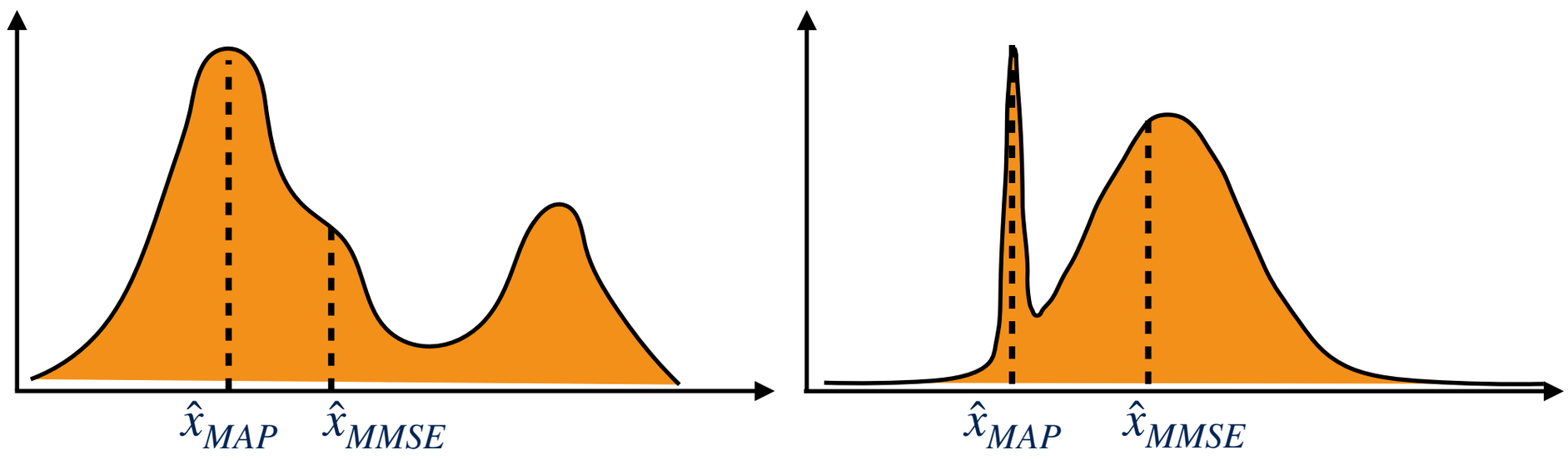
$$\hat{x}_{MAP} = \operatorname{argmax}_x \; p_{X|Y}(x \,|\, y) = \arg\min_{x \in \mathbb{R}^d} F(x, y) + U(x)$$

2. **MMSE estimation** (Monte-Carlo with samples from $p_{X|Y}$)

$$\hat{x}_{MMSE} = \arg\min_{u \in \mathbb{R}^d} \mathbb{E}_X[\|X - u\|^2 | \, Y = y] = \mathbb{E}_X[X \,|\, Y = y] = \int \tilde{x} \, p_{X|Y}(\tilde{x} \,|\, y) d\tilde{x}$$

Ex: if $N \sim$ i.i.d. std. Gaussian, $p_{Y|X}(y \,|\, x) \propto e^{-\|Ax-y\|^2/(2\sigma^2)}$, $p_X(x) \propto e^{-U(x)}$

$$\hat{x}_{MAP} = \arg\min_x \frac{1}{2\sigma^2} \|y - Ax\|_2^2 + U(x)$$



MAP estimation / optimisation

$$\operatorname*{argmin}_x F(x, y) + U(x)$$

→ Numerous sophistications of gradient descent to minimize $F + U$:
proximal gradient descent, primal-dual methods etc...

Known convergence properties in the convex case.

[Moreau 1965], [Combettes-Pesquet 2011], [Chambolle-Pock, 2011]...

Proximal gradient (Forward-Backward splitting)

$$x_{k+1} = \operatorname{prox}_{\varepsilon U}(x_k - \varepsilon \nabla F(x_k, y))$$

Plug-&-Play FBS

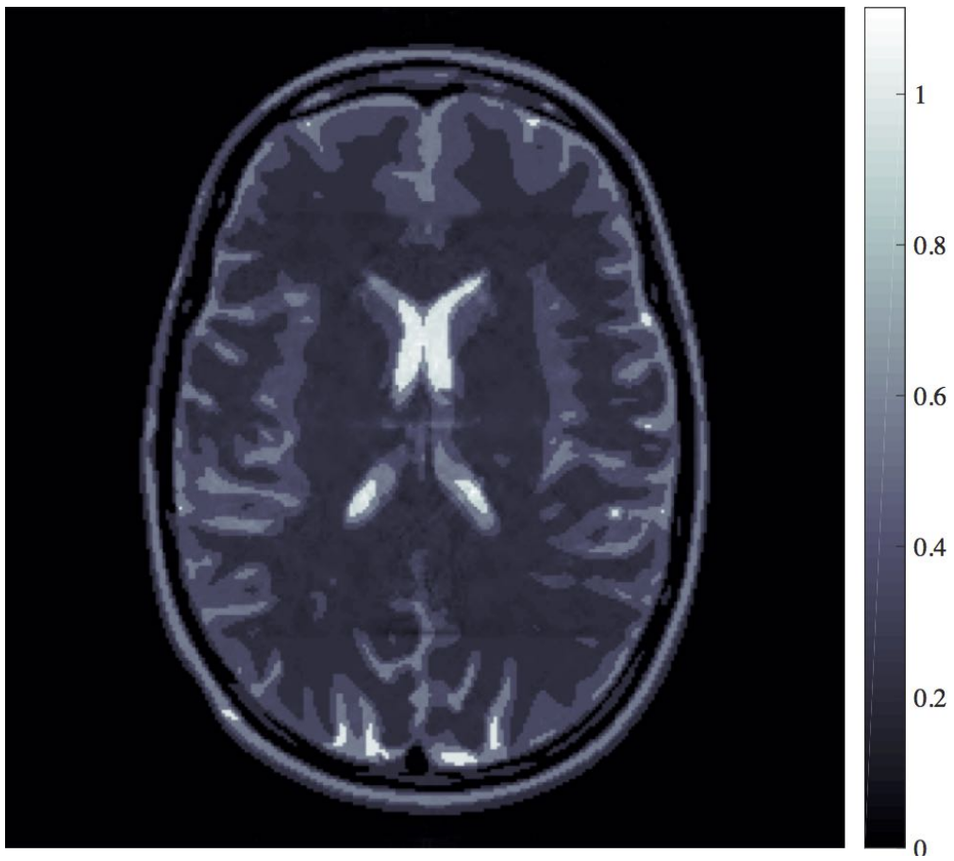
$$x_{k+1} = \quad D_\varepsilon \quad (x_k - \varepsilon \nabla F(x_k, y))$$

- Used with many proximal optim. schemes (primal-dual, ISTA, ADMM...)
- Fixed point convergence results under appropriate properties on D_ε and F.
- Convergence to stationary points of a functional for specifically trained denoisers.

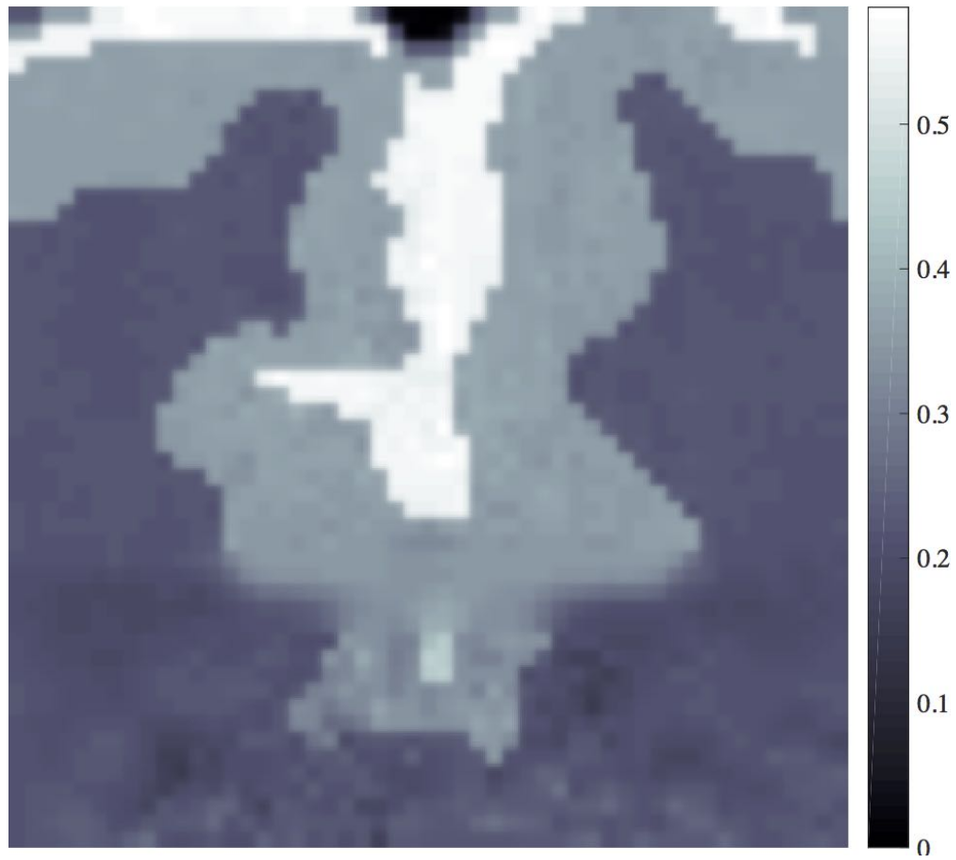
Sampling the posterior $p_{X|Y}$

Sampling the posterior $p_{X|Y}$ is necessary

- to compute the MMSE, or other pointwise estimators
- for quantifying uncertainties
- for more involved inferences, $p(X \in A \,|\, y)$, hypothesis testing, etc...



MRI for brain lesion



zoom

[Repetti et al. 2019]

Sampling $p_{X|Y}$?

Plug-&-Play sampling?

Sampling the posterior - Metropolis

Idea: build a **Markov Chain** (X_n) whose stationary measure is $p_{X|Y}$.

→ **Ergodic theorem:** if (X_n) is irreducible (countable state space) / Harris-recurrent (cont. state space) with stationary measure $p_{X|Y}$, then

$$\forall f \in L^1(p_{X|Y}), \qquad \frac{1}{n}\sum_{k=1}^{n} f(X_k) \underset{n\to\infty}{\longrightarrow} \int f(x) dp_{X|Y}(x) \text{ a.s.}$$

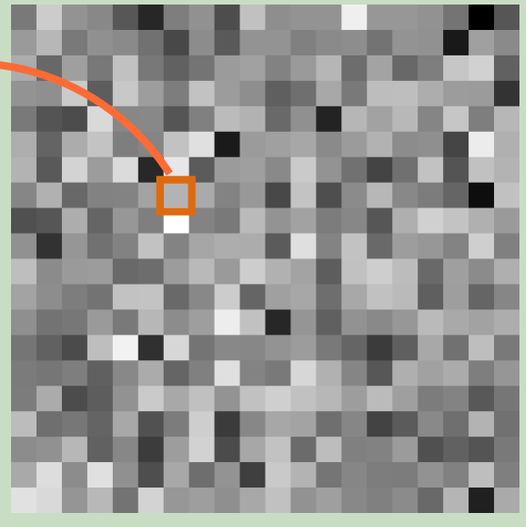
→ Central limit theorem with $O(\sqrt{n})$ rate under appropriate hypotheses

Bayesian uncertainty quantification, $p(X \in A \,|\, y)$ (hypothesis tests) etc...

Metropolis Algorithm for $p_{X|Y}(x) \propto e^{-H(x)}$ [Louchet, Moisan 2008]

Starting from a random image x_0 on the grid Ω, **for k from 0 to N**

1. $(i,j) \sim \mathcal{U}(\Omega)$, $z \sim \mathcal{U}([0,1])$
2. $x_{tmp} \leftarrow$ add $\pm t \sim \mathcal{U}([-\alpha,\alpha])$ to $x_k(i,j)$
3. $$x_{k+1} = \begin{cases} x_{tmp} & \text{if } z < e^{-(H(x_{tmp})-H(x_k))} \\ x_k & \text{otherwise} \end{cases}$$



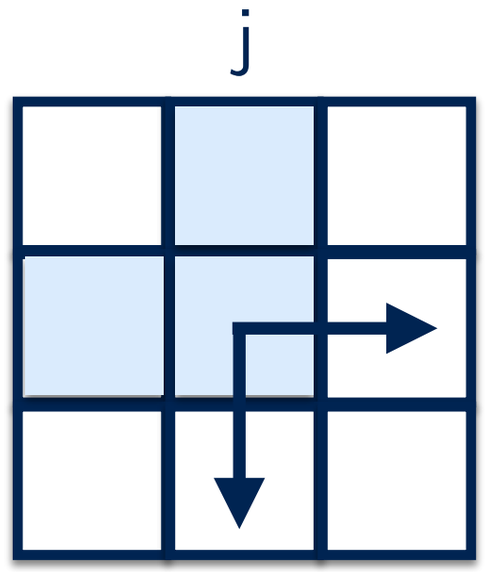
Sampling $e^{-\lambda TV(x)-\eta\|x\|^2}$ (with small η)

Efficient only if $H(x_{tmp}) - H(x_k)$ can be computed with local operations

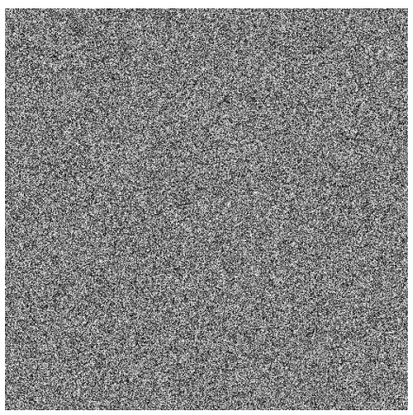
⟶ **parallelization on independent pixels.**

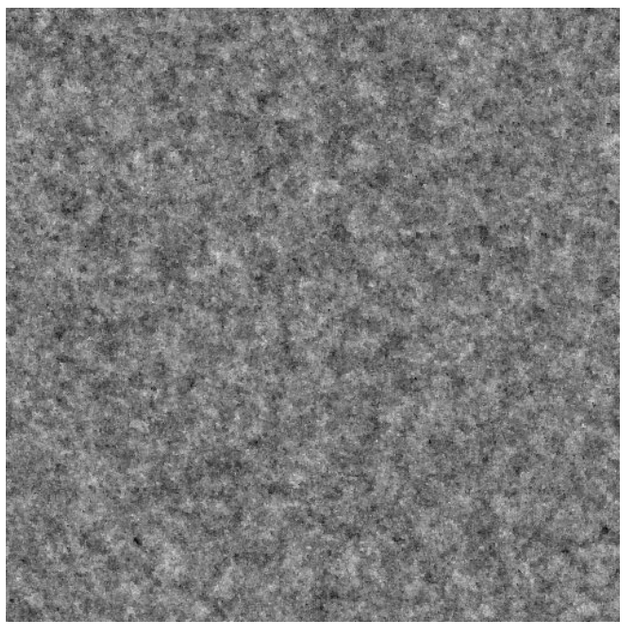
Ex: $H(x) = TV(x) + \eta\|x\|^2 = \sum_{i,j} (\|\nabla x\|_{i,j} + \eta\|x(i,j)\|^2)$

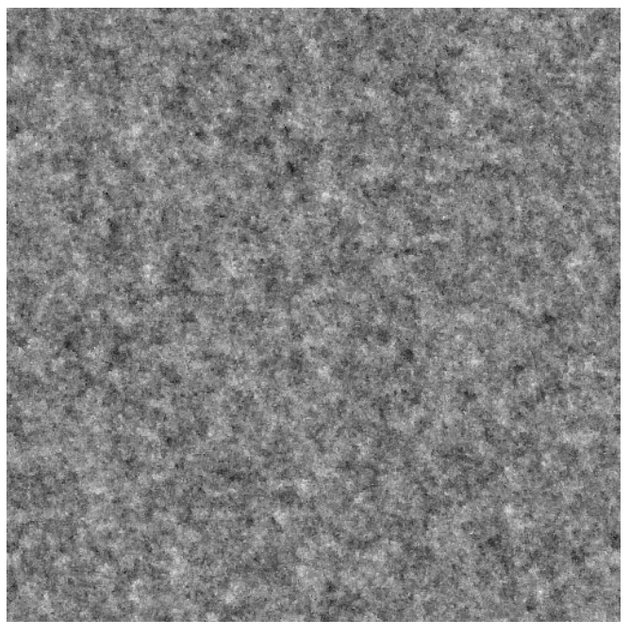
- Simple TV scheme
- Changing $x_k(i,j)$ only affects 3 terms in the sum
- $H(x_{tmp}) - H(x_k)$ only relies on $x_k(i,j)$ and its 4 neighbors.
- Pixels from a subsampled grid can be modified in parallel.



Initialization



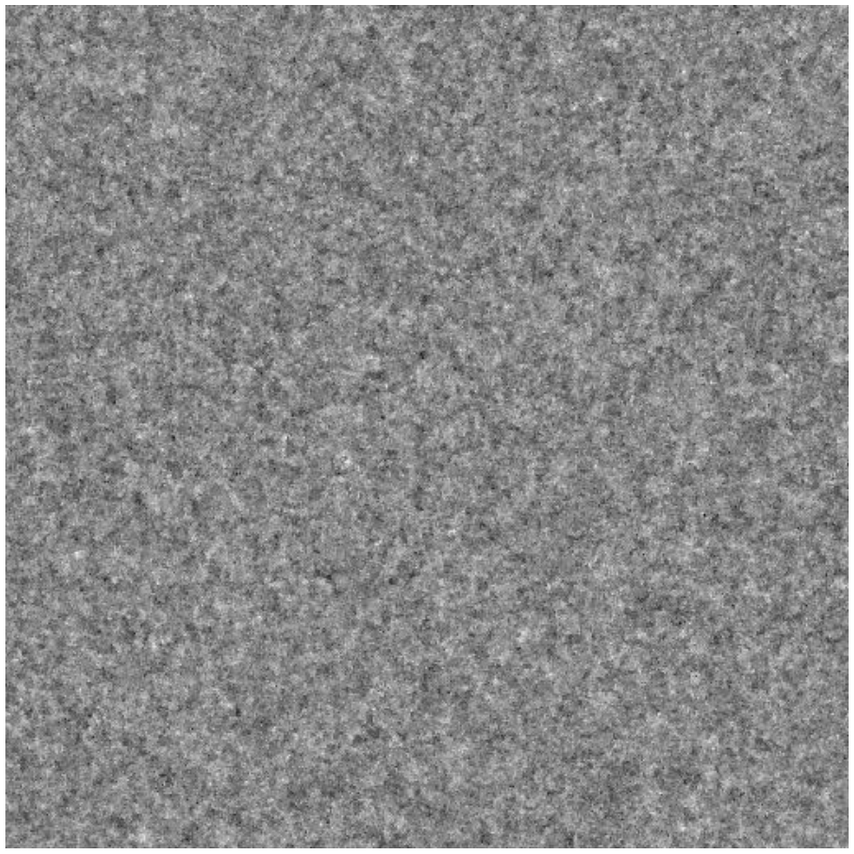




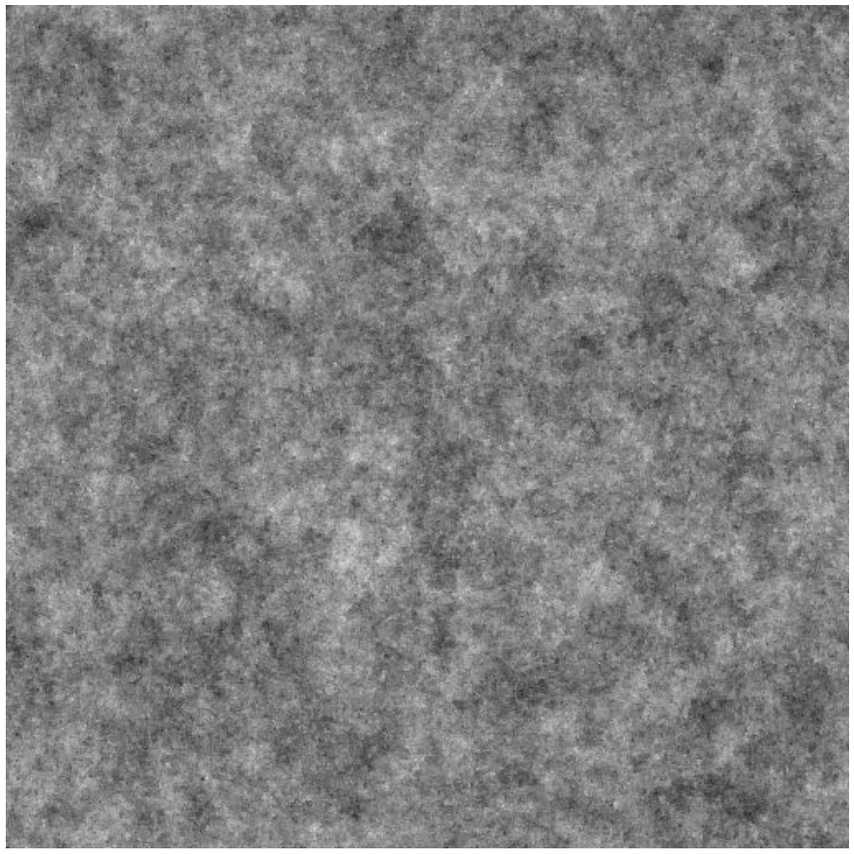
Two samples

Increasing λ

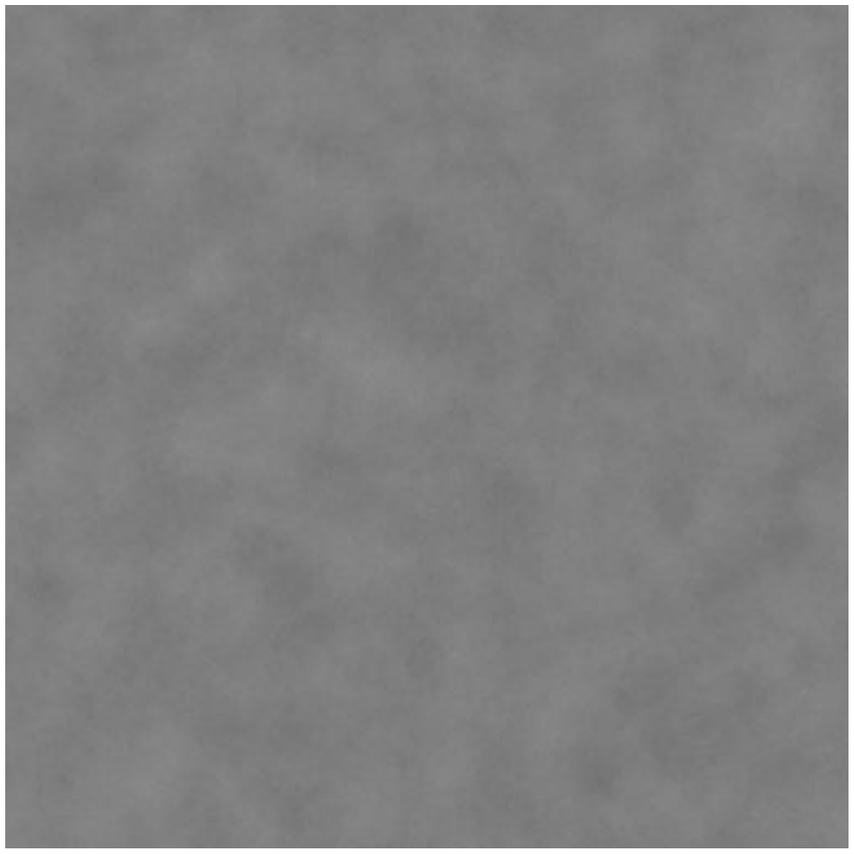
$\lambda = 1$



$\lambda = 10$



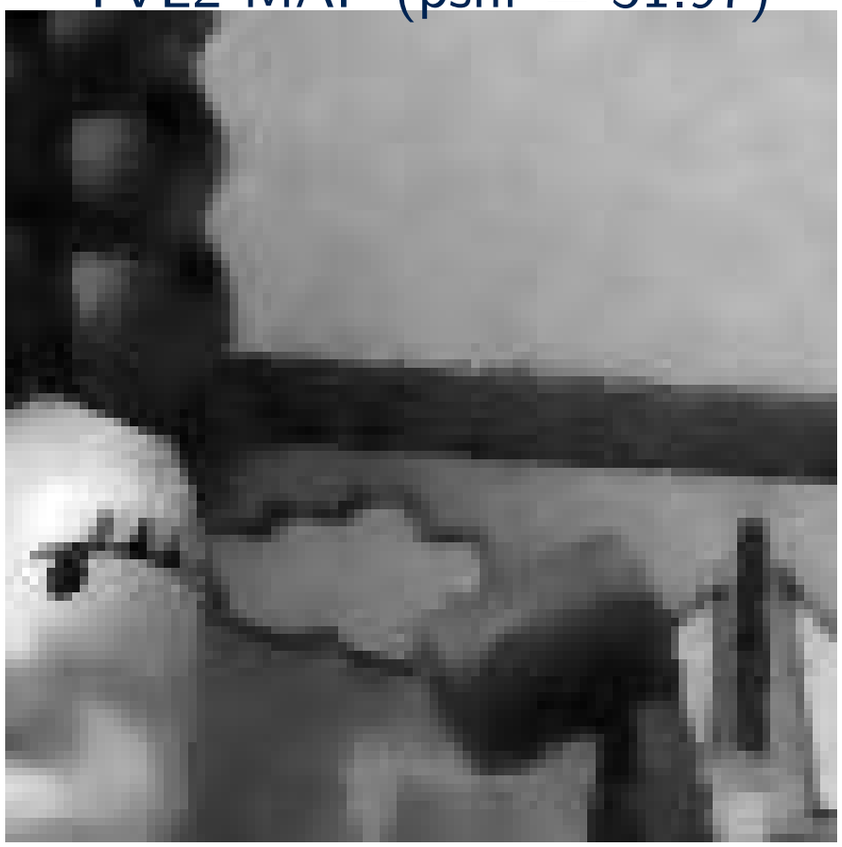
$\lambda = 100$

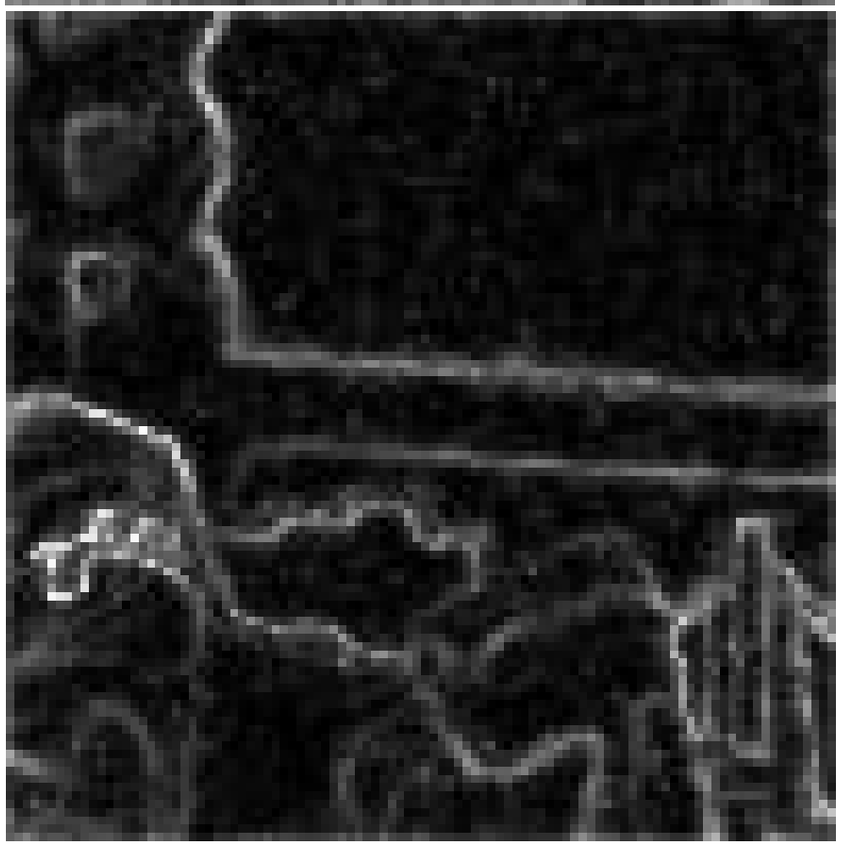


Sampling $e^{-\lambda TV(x) - \|x-y\|^2/2\sigma^2}$ for image denoising

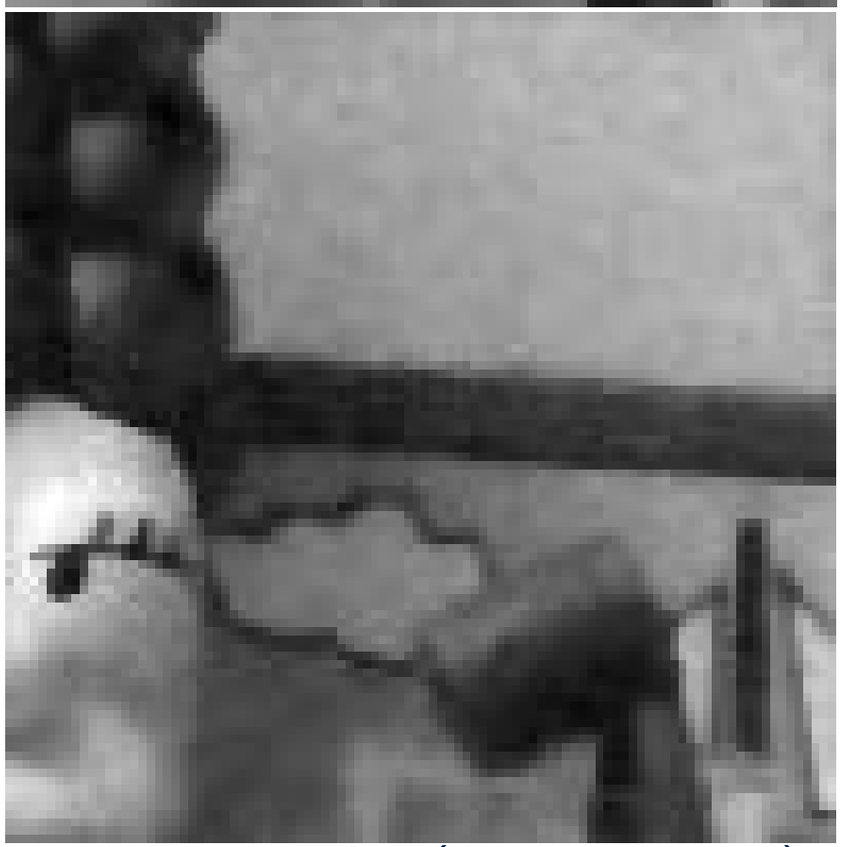
Observation y ($\sigma = 0.05$)

TVL2 MAP (psnr = 31.97)





TVL2 std



TVL2 MMSE (psnr = 31.56)

TV-LSE [Louchet, Moisan 2008]
No staircasing in the posterior average

Works well for sampling
$\propto e^{-\lambda TV(x) - \|Ax-y\|^2/2\sigma^2}$
with A working locally.

What if we want to sample from

$$\pi(u) \propto e^{-\lambda TV(u) - \|Au - u_0\|^2/2\sigma^2}$$

with A more complex or involving larger neighborhoods?

Unadjusted Langevin Algorithm (ULA)

Posterior **distribution** $\pi = p_{X|Y}(x\,|\,y) \propto e^{-H(x)}$

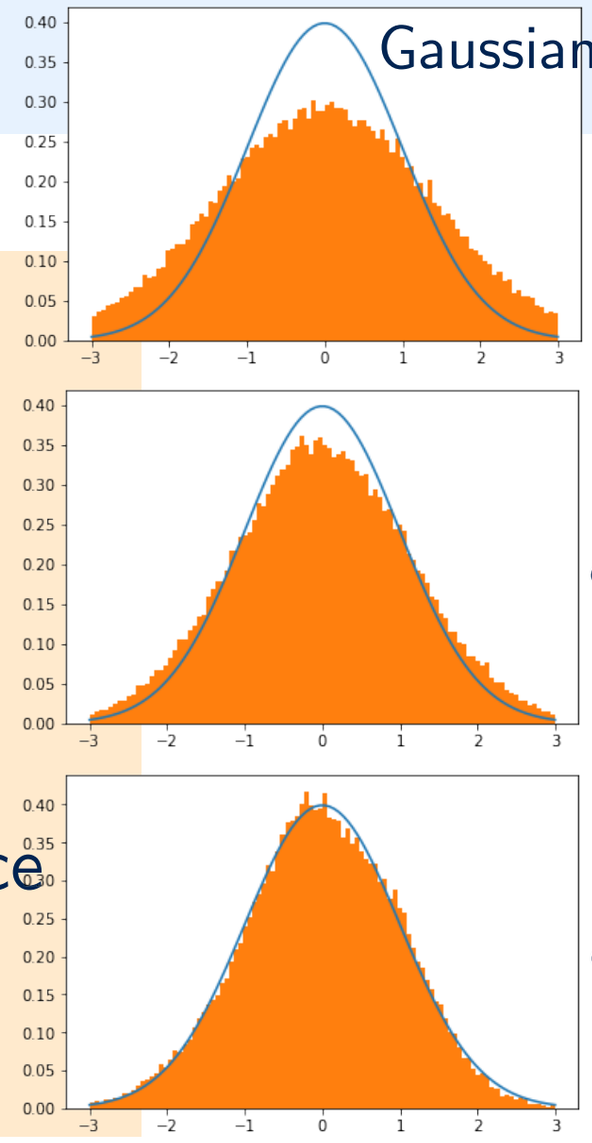
Unadjusted Langevin Algorithm

$$X_{k+1} = X_k - \delta \nabla H(X_k) + \sqrt{2\delta} Z_{k+1}$$

with $(Z_k)_{k\geq 0}$ i.i.d. $\mathcal{N}(0,\mathrm{Id})$

- Discretization of the Langevin SDE
- $(X_k)_{k\geq 0}$ homogeneous Markov Chain
- **Convergence** to unique stationary dist. $\pi_\delta \neq \pi$ if ∇H is L-Lipschitz and $\delta < 1/L$
- Exponentially fast **if** H strongly convex at ∞
- Theoretical results on the Wasserstein or TV distance between π_δ and π [Durmus, Moulines, 2017], [Durmus, De Bortoli, 2020]

Gaussian case



$\delta = 1$

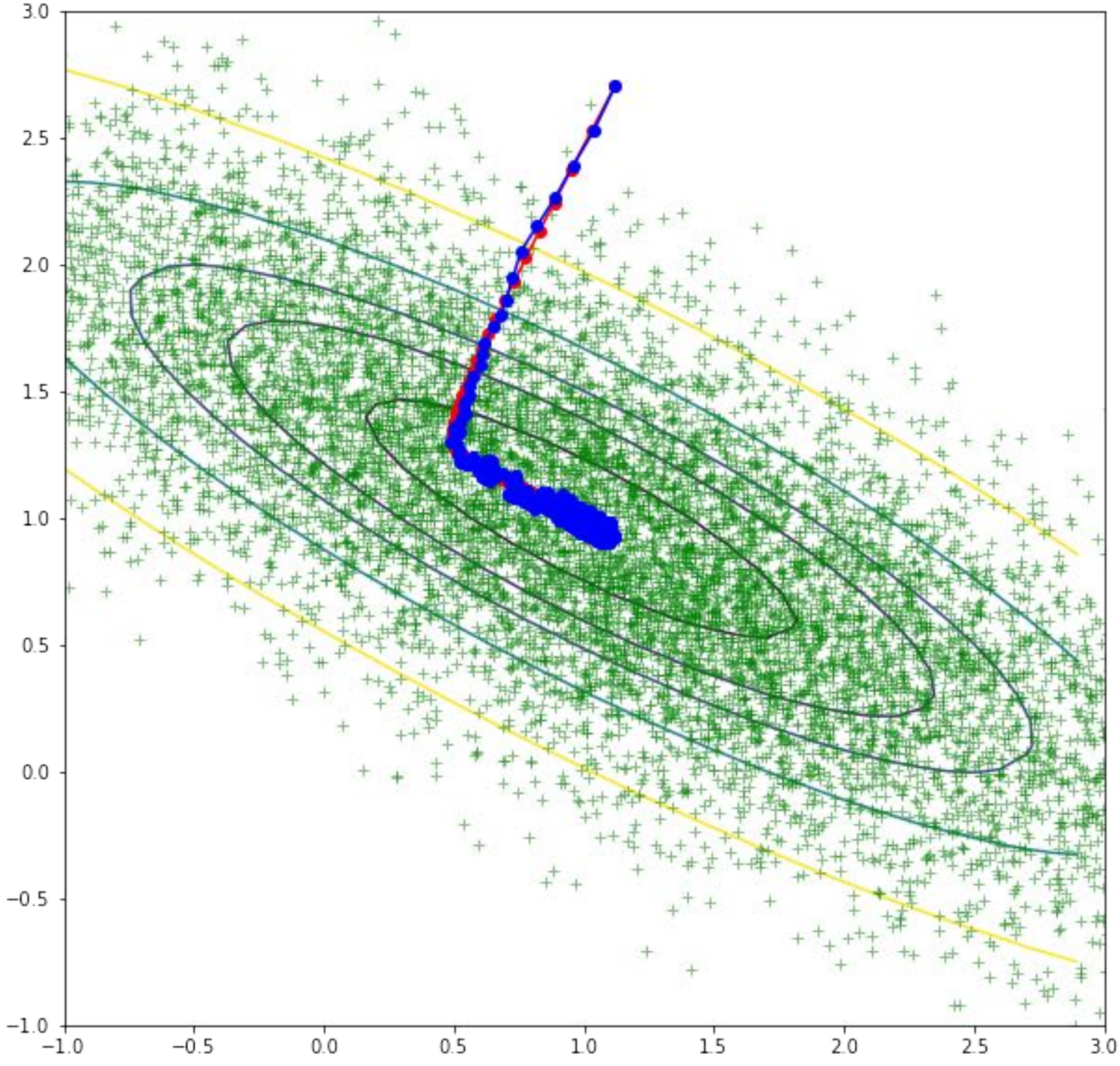
$\delta = 0.5$

$\delta = .01$

$\sigma_\delta^2 = 2/(2-\delta)$

$$p_X(x) \propto e^{-H(x)}$$

$$\text{(GD)} \quad X_{k+1} = X_k - \delta_k \nabla H(X_k) + \begin{cases} \delta_k Z_{k+1} \ \textbf{(SGD)} \\ \sqrt{2\delta_k} Z_{k+1} \ \textbf{(ULA)} \end{cases}$$



ULA for inverse problems in imaging

Target density

$$\pi(x) \propto e^{-H(x)} \text{ with } H(x) = F(x,y) + U(x)$$

with F and U both convex and smooth and grad-Lipschitz.

Unadjusted Langevin Algorithm

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}$$

with $(Z_k)_{k \geq 0}$ i.i.d. $\mathcal{N}(0, \mathrm{Id})$

regularised version of TV

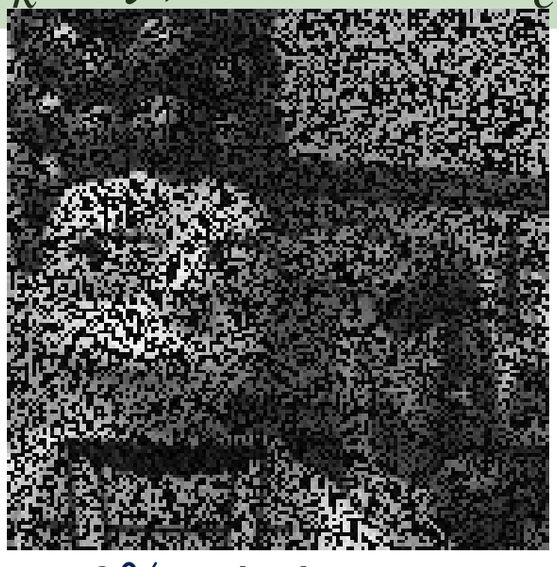
(Moreau-Yosida, or TV-Huber)

Exemple: $F(x,y) = \frac{1}{2\sigma^2}\|Ax - y\|_2^2$ and $U(x) = \lambda \mathrm{TV}_\epsilon(x)$

$$X_{k+1} = X_k - \delta A^*(AX_k - y) - \delta\lambda \nabla \mathrm{TV}_\epsilon(X_k) + \sqrt{2\delta} Z_{k+1}$$

MYULA [Durmus et al. 2018]:
Use Moreau-Yosida regularization of U

$$\nabla U^\lambda(x) = \frac{1}{\lambda}(x - \mathrm{prox}_{\lambda U}(x))$$



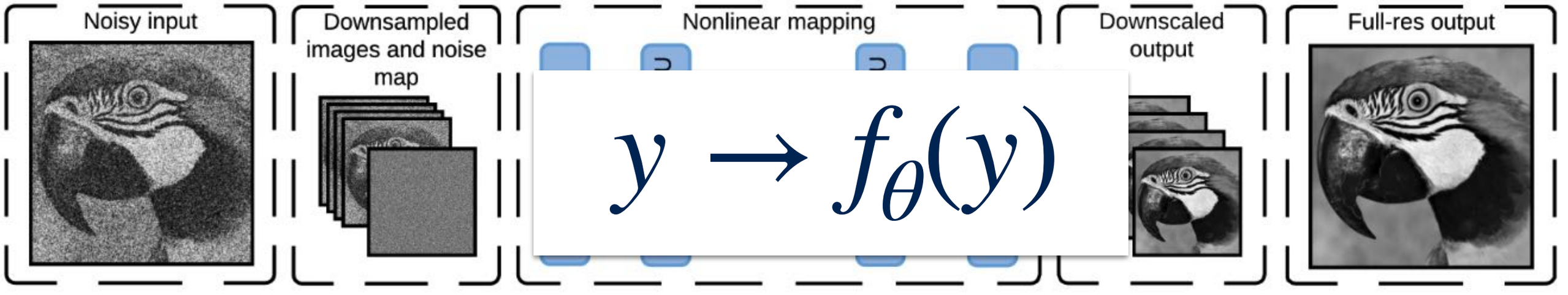
50% missing

reconstructed posterior

posterior

PnP sampling

End-to-end neural networks



Set $(x_i, y_i)_{i=1}^N$ of independent realizations of the degradation model

Training: $$\hat{\theta} = \arg\min_{\theta \in \Theta} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \mathscr{L}(f_\theta(y_i), x_i)}_{\simeq \mathbb{E}[\mathscr{L}(f_\theta(Y) - X)]}$$

If $\mathscr{L}(v, w) = \|v - w\|^2$
$f_{\hat{\theta}}(y) \simeq \mathbb{E}[X \,|\, Y = y]$

Limitations of the approximation of $\mathbb{E}[X \,|\, Y = y]$:

- Finite number of samples N
- Limited expressive power f_θ
- Optimization not perfect

Probably not so bad for simple inverse problems like denoising

Combining NN with the Bayesian framework

Denoising problem with Gaussian additive noise

$$Y = X + N$$

with $N \sim \mathcal{N}(0, \varepsilon\, \mathbb{I}_d)$ ($\sqrt{\varepsilon}$ = noise std) and $X \sim p_X \propto e^{-U}$

MMSE denoiser [Tweedie] [Efron et al. 2011]**:** if $D_\varepsilon^\star(x) := \mathbb{E}(X \,|\, Y = x)$, then

$$D_\varepsilon^\star(x) = x - \varepsilon \nabla U_\varepsilon(x)$$

with $U_\varepsilon = -\log(\underbrace{p_X * g_\varepsilon}_{p_\varepsilon})$ (g_ε Gaussian of variance ε).

Proof: notice that $p_{X|Y}(t, x) \propto g_\varepsilon(x - t) p_X(t)$

$$\nabla \log p_X * g_\varepsilon(x) = \frac{\int \nabla g_\varepsilon(x - t) p_X(dt)}{\int g_\varepsilon(x - t) p_X(dt)} = -\frac{1}{\varepsilon} \frac{\int (x - t) g_\varepsilon(x - t) p_X(dt)}{\int g_\varepsilon(x - t) p_X(dt)} = -\frac{1}{\varepsilon}(x - \mathbb{E}(X \,|\, Y = x))$$

An MMSE denoiser $D_\varepsilon = D_\varepsilon^\star$ can be plugged in any scheme using a **gradient step** on U to solve inverse problems with prior $\propto e^{-U}$.

Makes plug-and-play possible for optimization and sampling schemes relying on the gradient of the log-prior (gradient desc., ULA)

PnP sampling

Goal : sample the posterior $p_{X|Y}(x\,|\,y) = e^{-U(x)-F(x,y)}$

PnP- ULA:

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) \underbrace{- \delta \nabla U(X_k)}_{\frac{\delta}{\varepsilon}(X_k - D_\varepsilon(X_k))} + \sqrt{2\delta} Z_{k+1}$$

with $(Z_k)_{k\geq 0}$ i.i.d. $\mathcal{N}(0,\mathrm{Id})$

Idea appears in [Alain and Bengio 2014]. Similar ideas in [Guo et al. 2019], [Kadkhodaie et al. 2020], [Kawar et al. 2020], [Bigdeli et al. 2020]...
First convergence analysis in [Laumont et al. 2021].

Implicit representation: the true MMSE denoiser $D_\varepsilon^\star$ is not known. PnP methods rely on denoisers D_ε assumed to be a good approximation of it.
Make sense for CNN denoisers (trained to minimize the quadratic risk).

Versatility/flexibility: same advantages as proximal PnP

PnP-ULA convergence

Oracle PnP-ULA [Laumont et al. 2021]

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \frac{(X_k - D_\varepsilon^\star(X_k))}{\varepsilon} - \delta \frac{X_k - \Pi_C(X_k)}{\lambda} + \sqrt{2\delta} Z_{k+1}$$

with $(Z_k)_{k\geq 0}$ i.i.d. $\mathcal{N}(0,\mathrm{Id})$

projection on $B(0,R_C)$
ensures geometrically fast convergence

Hypotheses for convergence

- Likelihood $p_{Y|X}$ bounded, C^1 and $\nabla \log p_{Y|X}$ Lipschitz
- MSE loss is finite and uniformly bounded for the denoising problem under p_ε

→ posterior $p_\varepsilon(x \,|\, y) \propto p_{Y|X}\, p_\varepsilon$ is smooth and ∇-Lipschitz, and can be made arbitrary close to $p_{X|Y}$

→ MMSE well defined

PnP-ULA convergence

PnP-ULA [Laumont et al. 2021]

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \frac{(X_k - D_\varepsilon(X_k))}{\varepsilon} - \delta \frac{X_k - \Pi_C(X_k)}{\lambda} + \sqrt{2\delta} Z_{k+1}$$

with $(Z_k)_{k\geq 0}$ i.i.d. $\mathcal{N}(0,\mathrm{Id})$

Hypotheses on the denoiser D_ε

- D_ε is Lipschitz (true for CNN denoisers)
- $\exists M : \mathbb{R}^+ \to \mathbb{R}^+$, $\forall x$ s.t. $\|x\| \leq R$, $\|D_\varepsilon(x) - D_\varepsilon^\star(x)\| \leq M(R)$.

Non asymptotic error when sampling [Laumont et al., 2021]: if $\varepsilon, \lambda, \gamma$ small enough

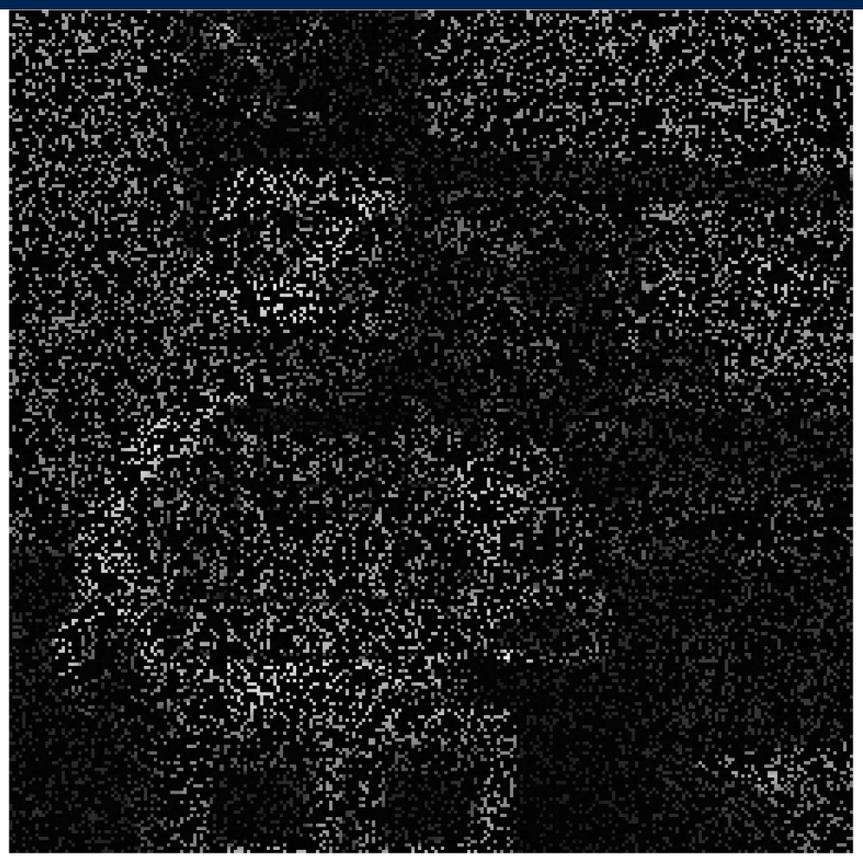
$$\left| \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}[h(X_k)] - \int_{\mathbb{R}^d} h(\tilde{x}) p_\varepsilon(\tilde{x} \,|\, y) d\tilde{x} \right| \leq C_0 \left(\frac{C_1}{R_C} + C_2 \left(\sqrt{\delta} + \frac{1}{n\delta} + C_M \right) \right)$$

R_C convex size

step δ

constant depending on $M(R)$

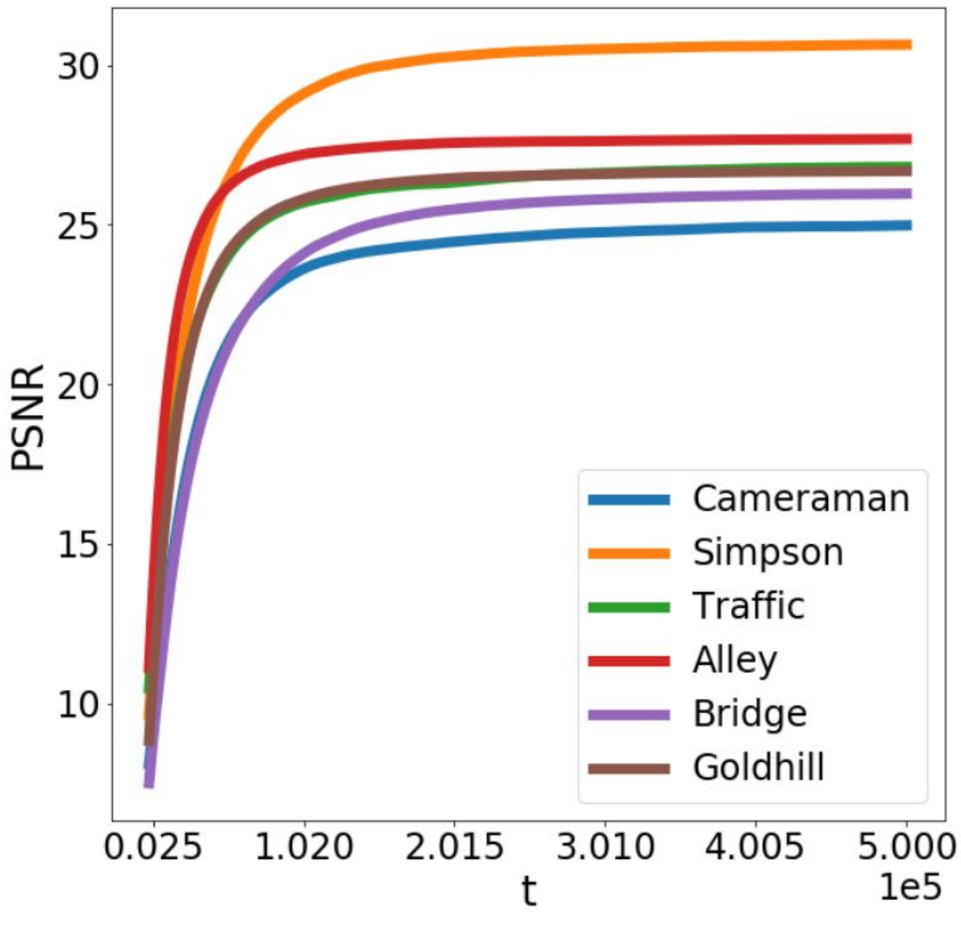
PnP-ULA for inpainting



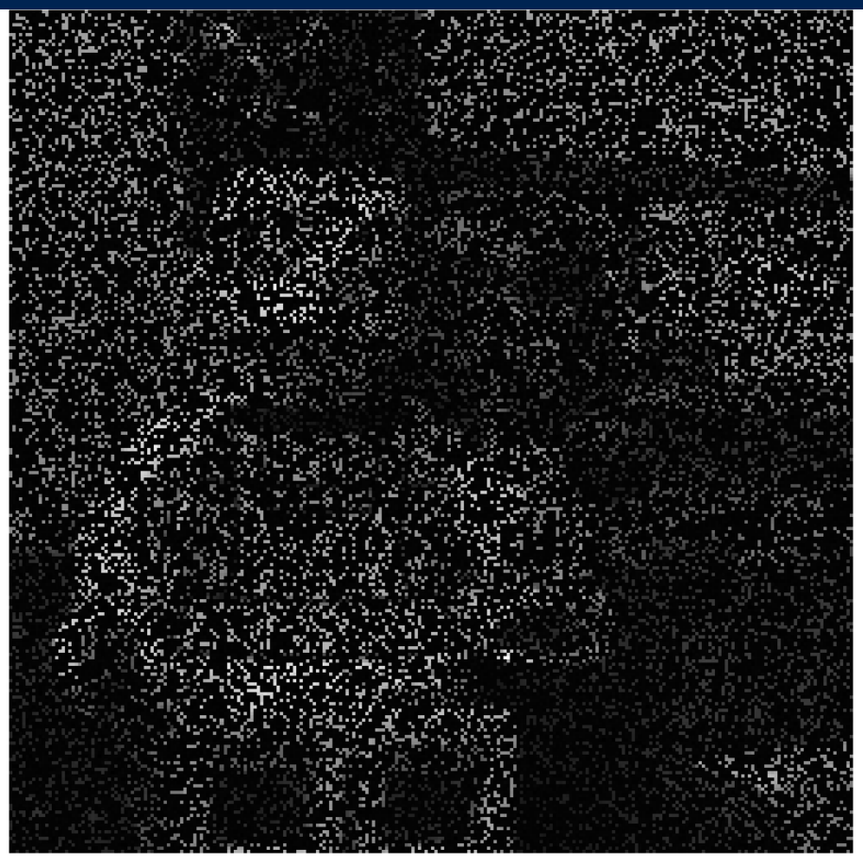
80% missing pixels

D_ε = SN-DnCNN provided by [Ryu et al., 2019], trained s.t. $Id - D_\varepsilon$ is L-Lipschitz for $L = 1$.

$\varepsilon = (5/255)^2$

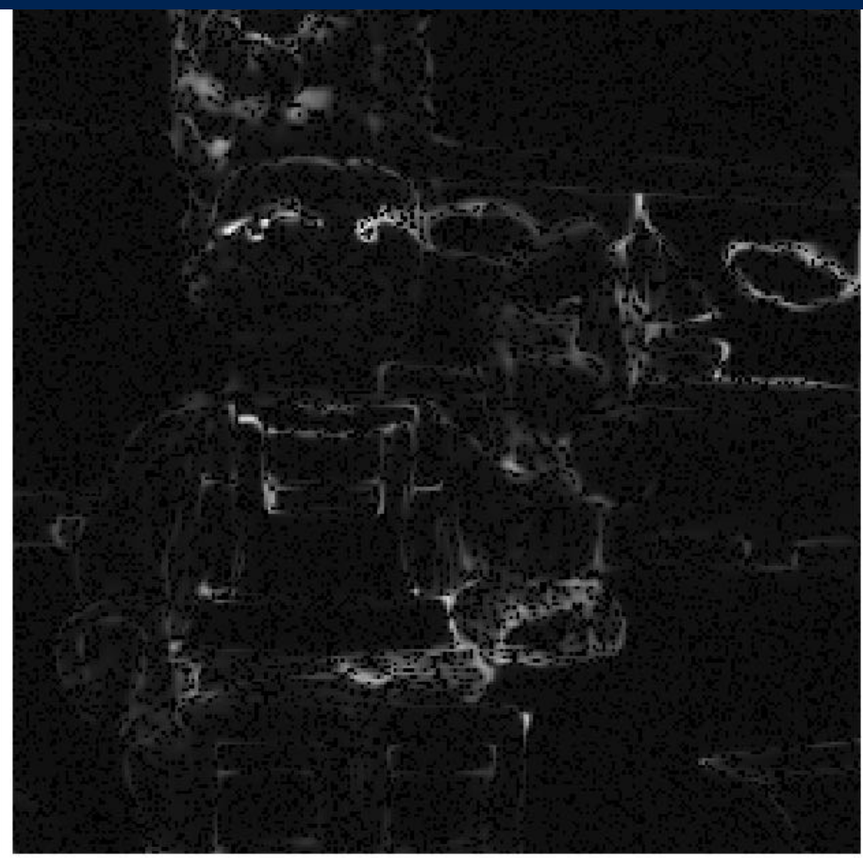


PnP-ULA for inpainting



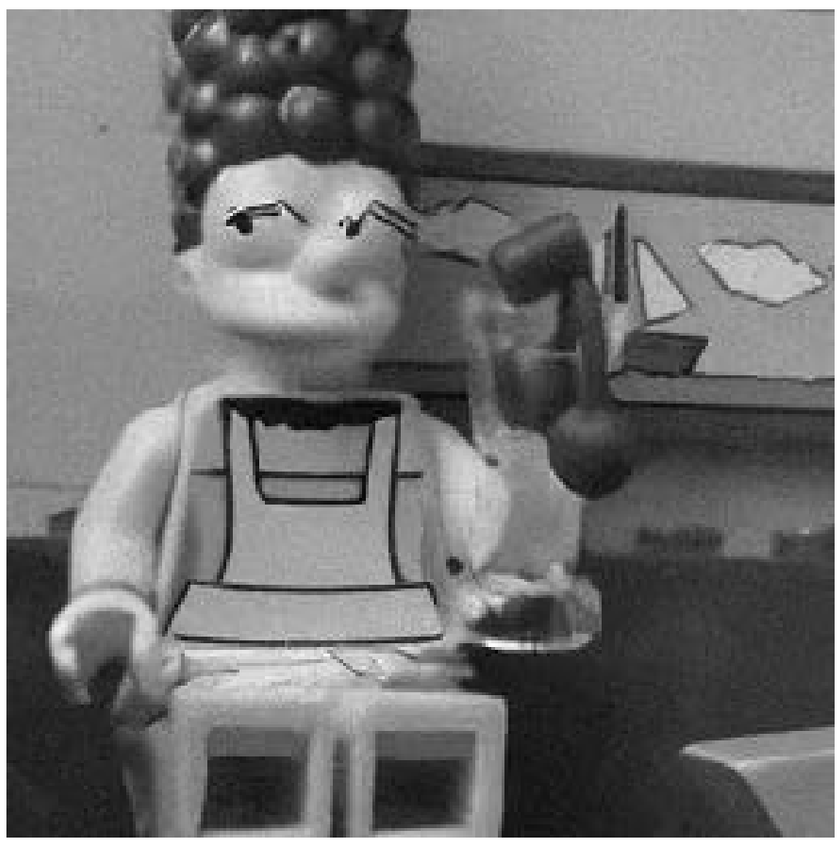
80% missing pixels

PnP-ULA posterior mean



posterior std

MAP

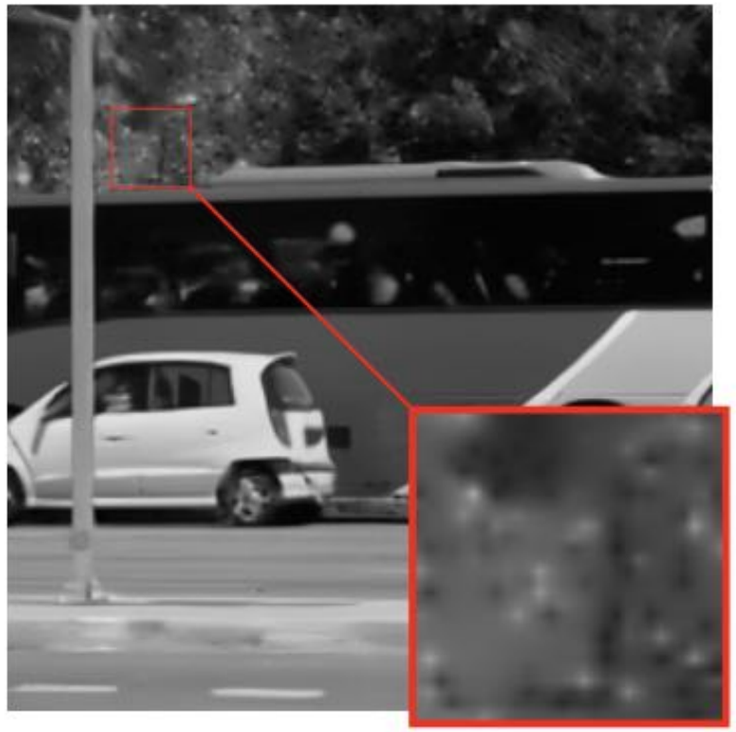


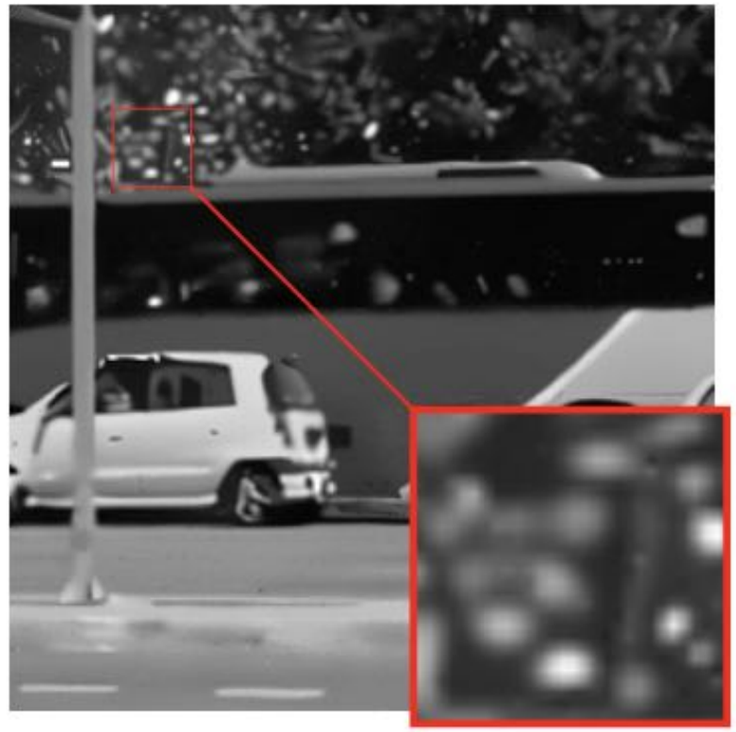
samples

PnP-ULA for inpainting

MMSE

MAP





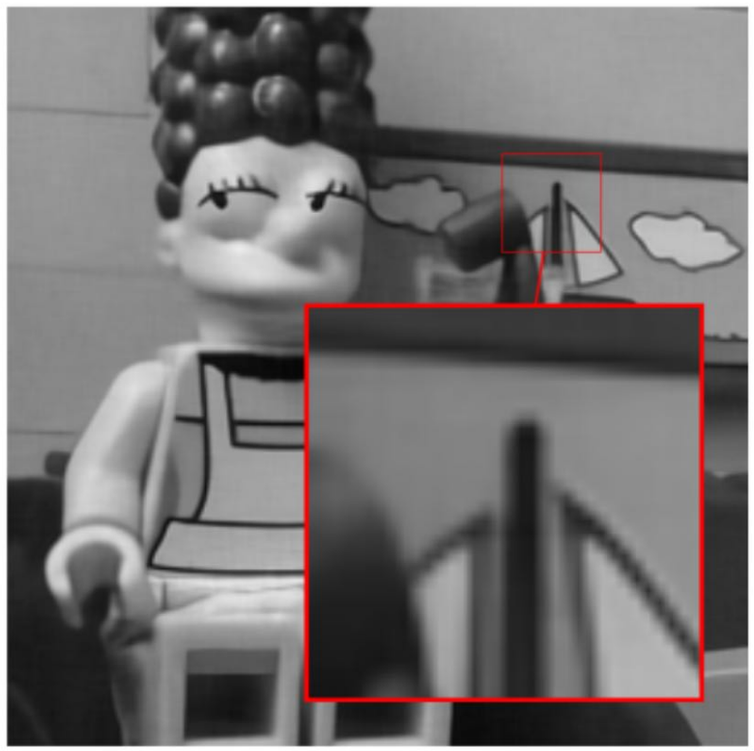
PnP-ULA for 9x9 deblurring

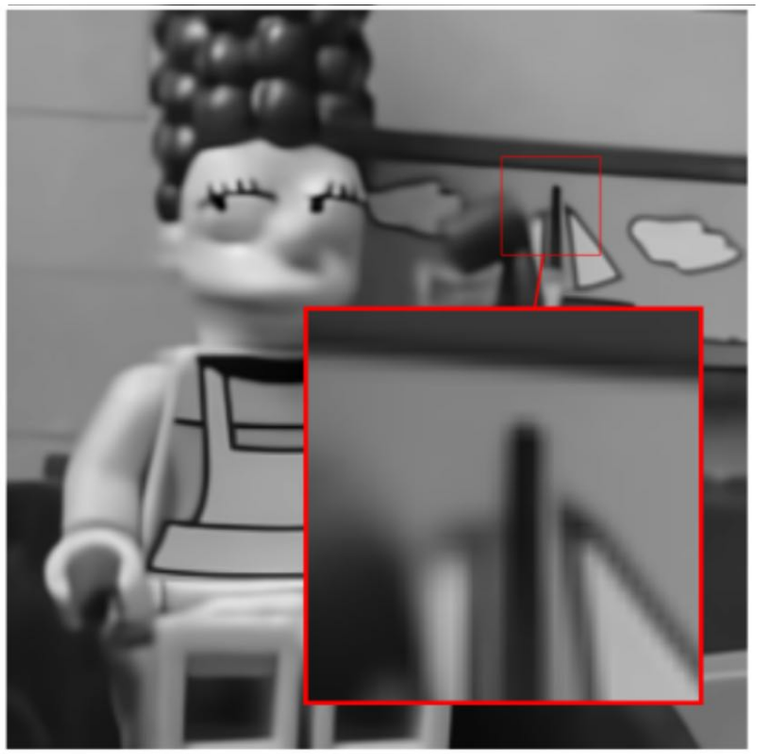
[Laumont et al., 2021]

PnP-ULA for 9x9 deblurring

MMSE

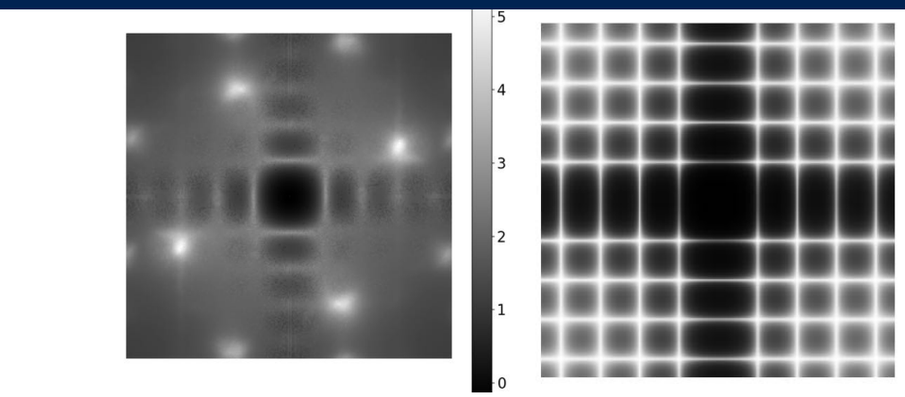
MAP





[Laumont et al., 2021]

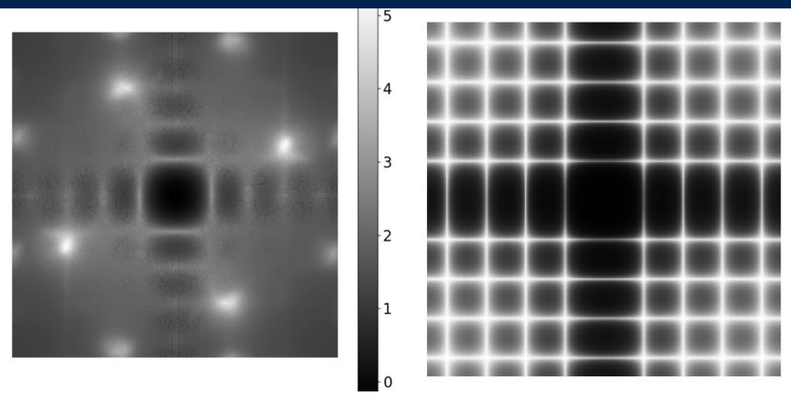
Are denoisers appropriate image priors?



PnP-Deblurring, 9x9 uniform kernel

SN-DnCNN MMSE

Are denoisers appropriate image priors?



PnP-Deblurring, 9x9 uniform kernel

SN-DnCNN MMSE

FINE MMSE
[Pesquet et al., 2020]