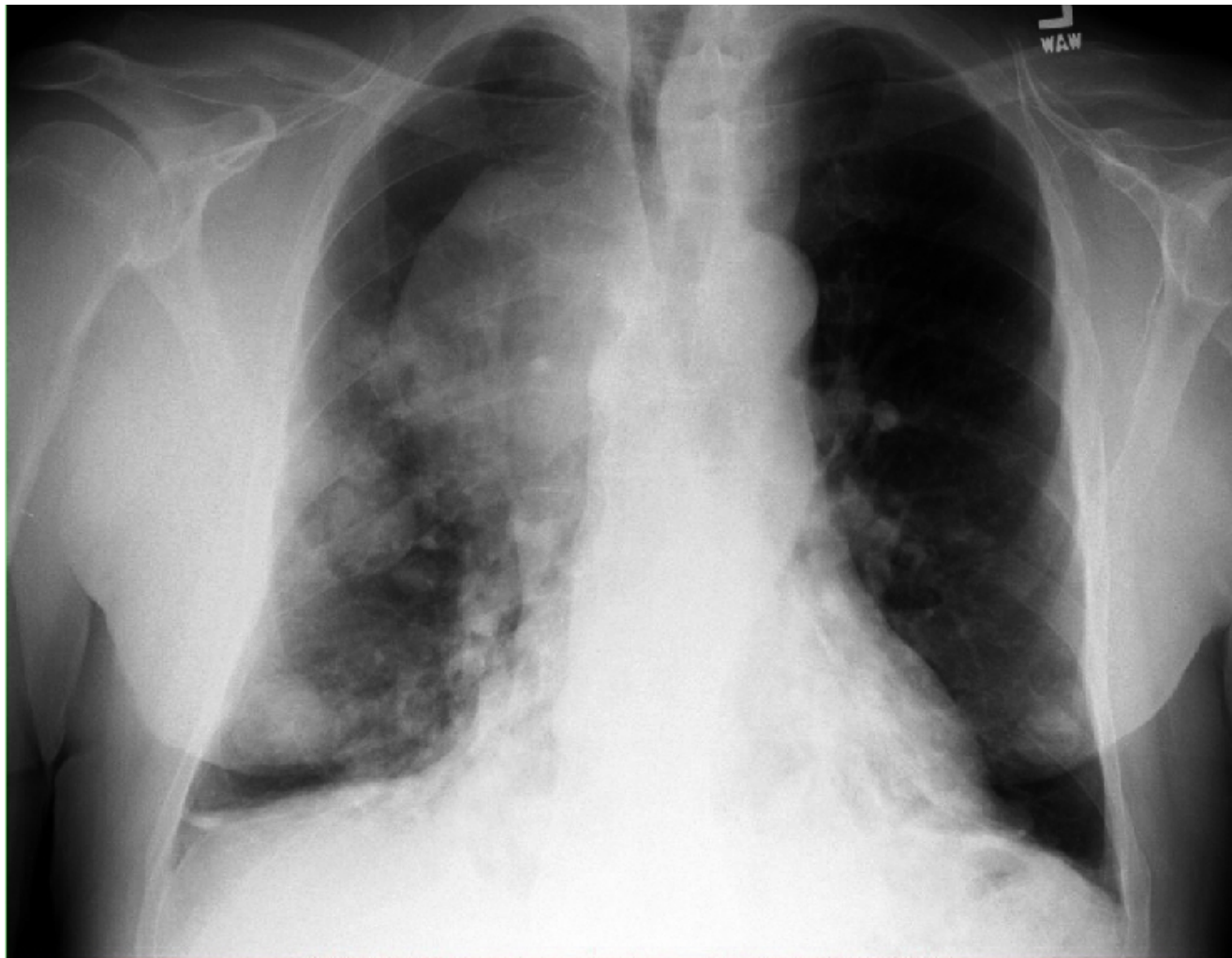


Image restoration

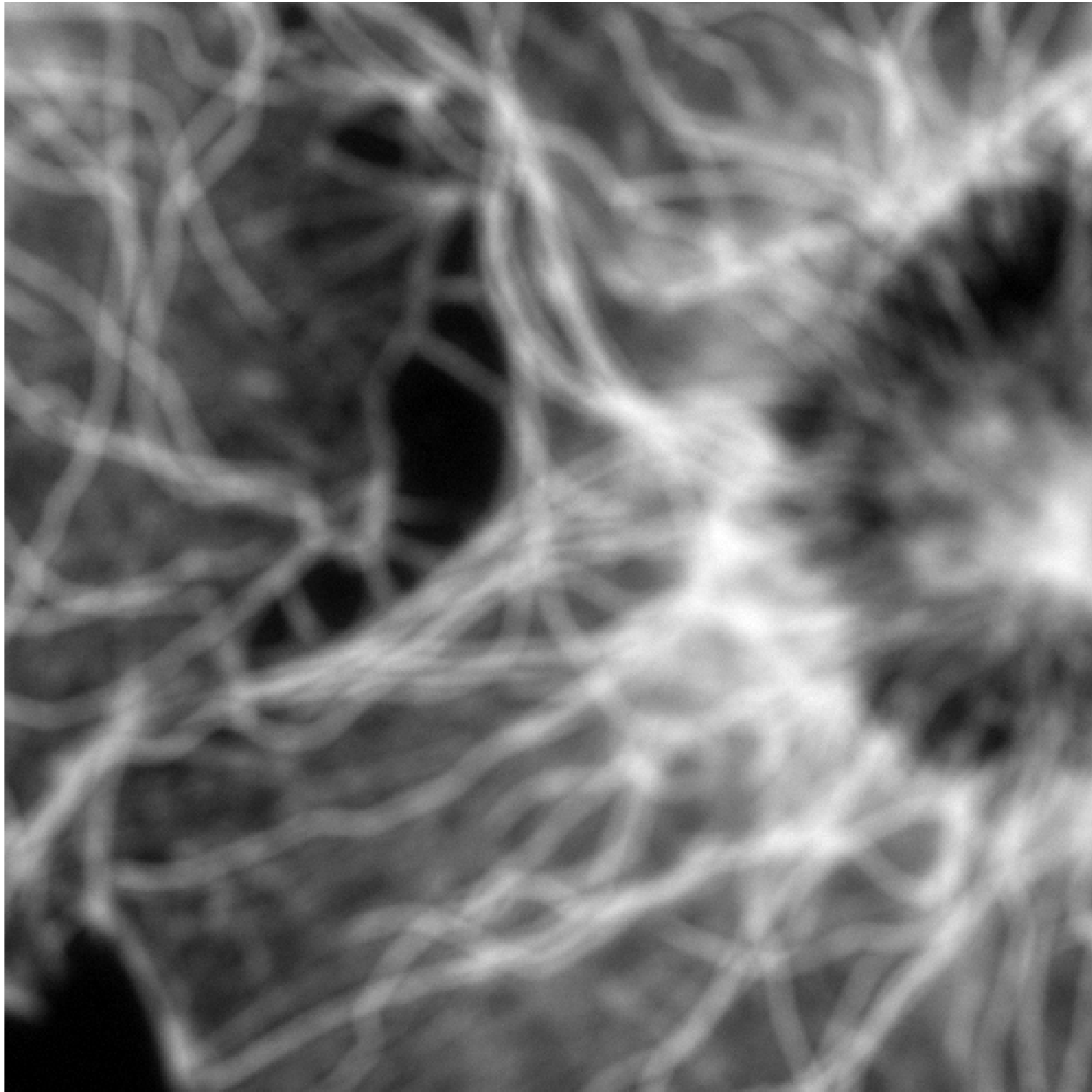
Julie Delon
M2 MVA

Denoising



- Noise sources: *shot* noise, readout noise, transmission errors, etc

Deblurring



R. Abergel, L. Moisan

Deblurring



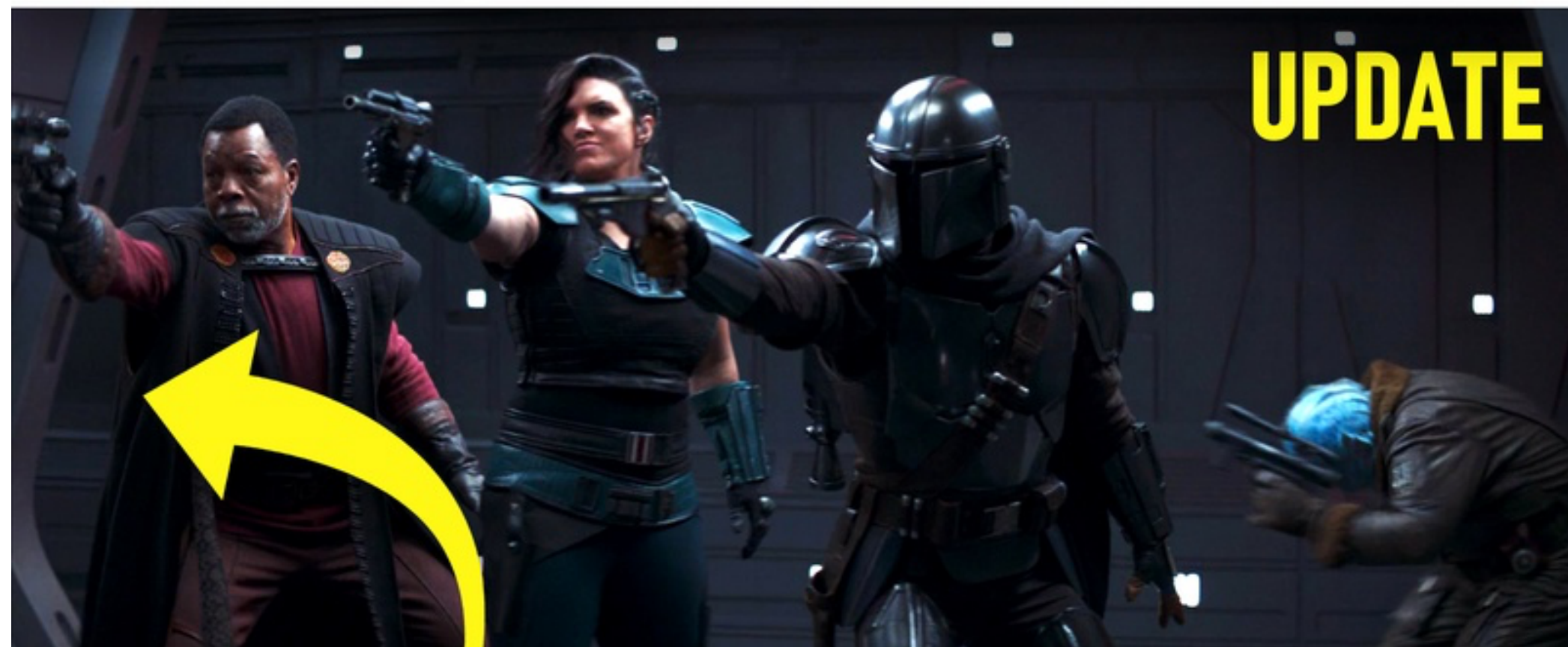
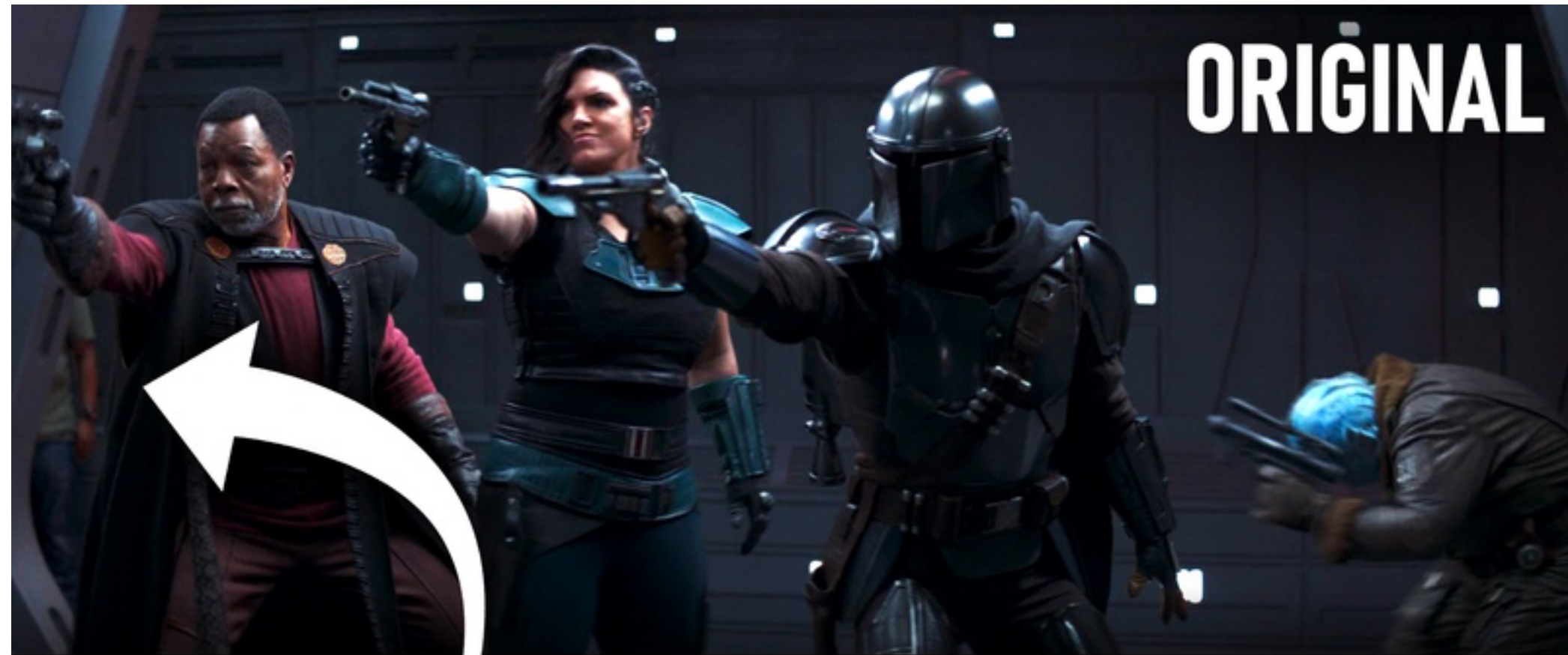
Missing pixels



Missing pixels



Inpainting



[The Mandalorian, Disney, 2020]

Inpainting

séquence modifiée



séquence originale



zone générée



Inpainting

séquence modifiée



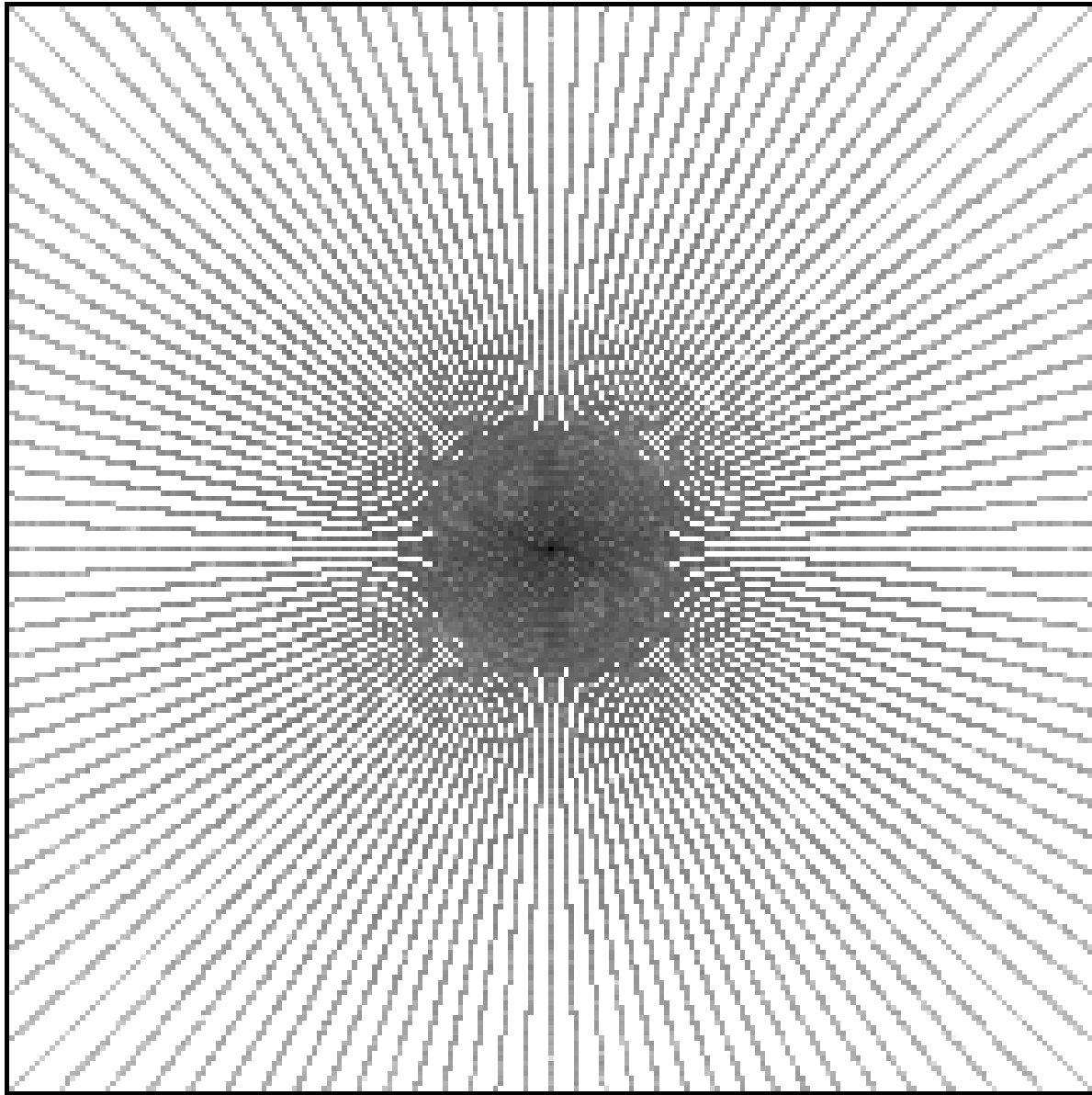
séquence originale



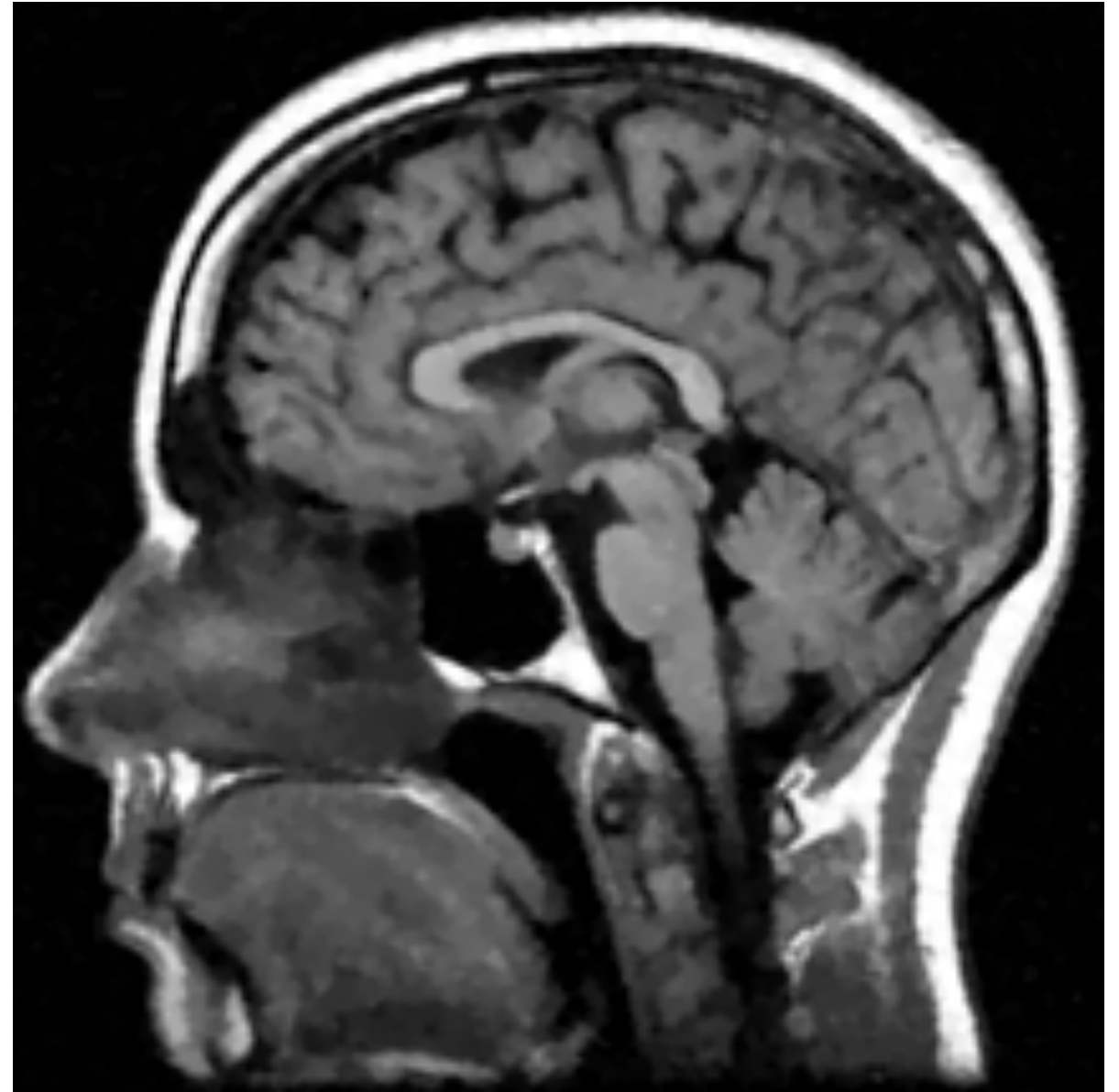
zone générée



MRI



measure



reconstruction

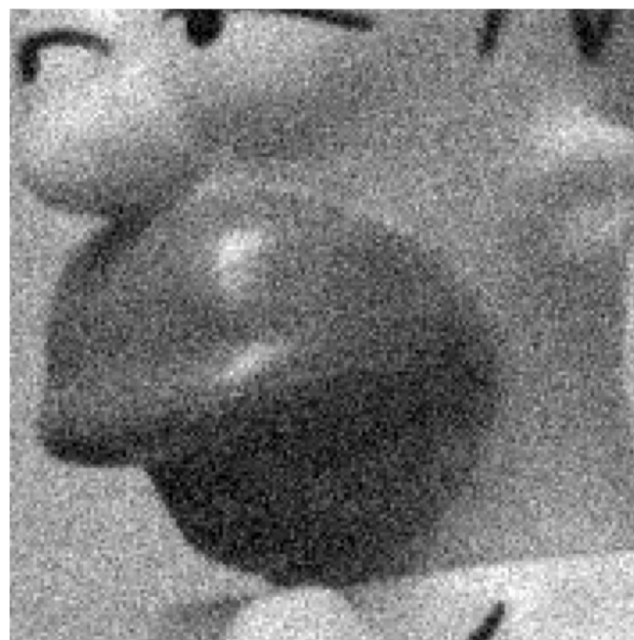
Images : R. Abergel, L. Moisan

Inverse problems

$$\begin{array}{ccccccc} & \mathbb{R}^d & & \mathbb{R}^{d \times n} & & \mathbb{R}^n & & \mathbb{R}^d \\ & & & & & & & \\ \nearrow & y & = & Ax & + & n & \nwarrow \\ \text{observation} & & & \text{degradation operator} & & \text{unknown} & & \text{noise} \end{array}$$

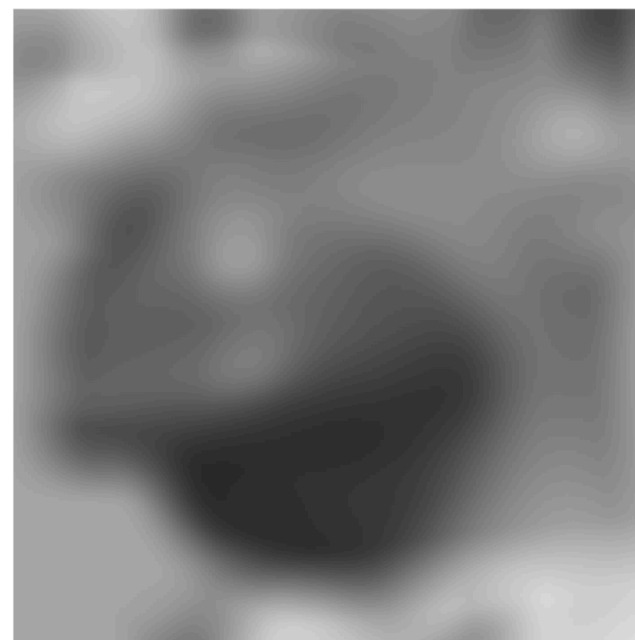


image x



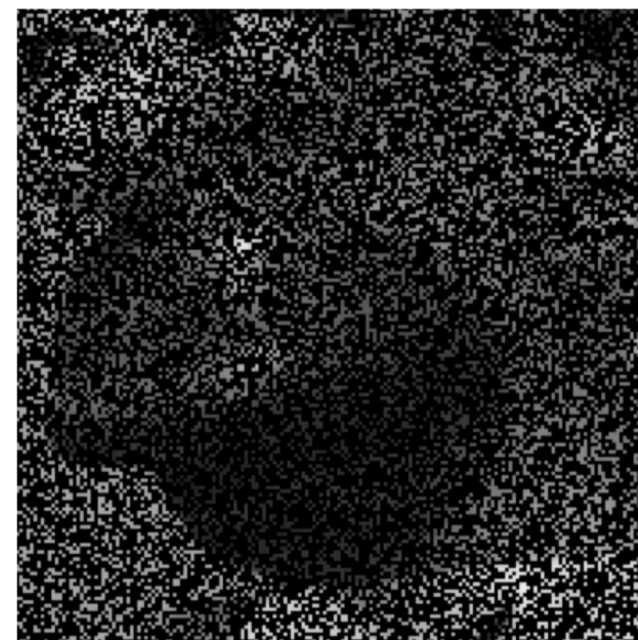
additive noise

$$A = I_n$$



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

Classical approaches

- Least squares

$$\hat{x} = (A^*A)^{-1}A^*y$$



not always one-to-one
and its eigenvalues
may be small.



A poorly conditioned or rank deficient!

Classical approaches

- Least squares

$$\hat{x} = (A^*A)^{-1}A^*y$$


not always one-to-one
and its eigenvalues
may be small.

A poorly conditioned or rank deficient!

Classical approaches

- Least squares

$$\hat{x} = (A^*A)^{-1}A^*y$$



not always one-to-one
and its eigenvalues
may be small.

- Tikhonov regularization

$$\begin{aligned}\hat{x} &= \operatorname{argmin}_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2 \\ &= (A^*A + \lambda I)^{-1}A^*y\end{aligned}$$



A poorly conditioned or rank deficient!

Bayesian formalism in image processing

JOURNAL OF THE OPTICAL SOCIETY OF AMERICA

VOLUME 62, NUMBER 1

JANUARY 1972

Bayesian-Based Iterative Method of Image Restoration*

WILLIAM HADLEY RICHARDSON

Visibility Laboratory, University of California, San Diego, San Diego, California 92152

(Received 15 September 1970)

An iterative method of restoring degraded images was developed by treating images, point spread functions, and degraded images as probability-frequency functions and by applying Bayes's theorem. The method functions effectively in the presence of noise and is adaptable to computer operation.

INDEX HEADINGS: Spread function; Image restoration; Deconvolution.

IEEE TRANSACTIONS ON COMPUTERS, VOL. C-26, NO. 3, MARCH 1977

219

Bayesian Methods in Nonlinear Digital Image Restoration

B. R. HUNT

Abstract—Prior techniques in digital image restoration have assumed linear relations between the original blurred image intensity, the silver density recorded on film, and the film-grain noise. In this paper a model is used which explicitly includes nonlinear

virtue of an intensity distribution of some radiated energy (e.g., light-waves, penetrating radiation). The radiated energy is acted upon by an image formation system (e.g., optical lenses or opaque pinhole collimators for pene-

Bayesian framework

Forward degradation model

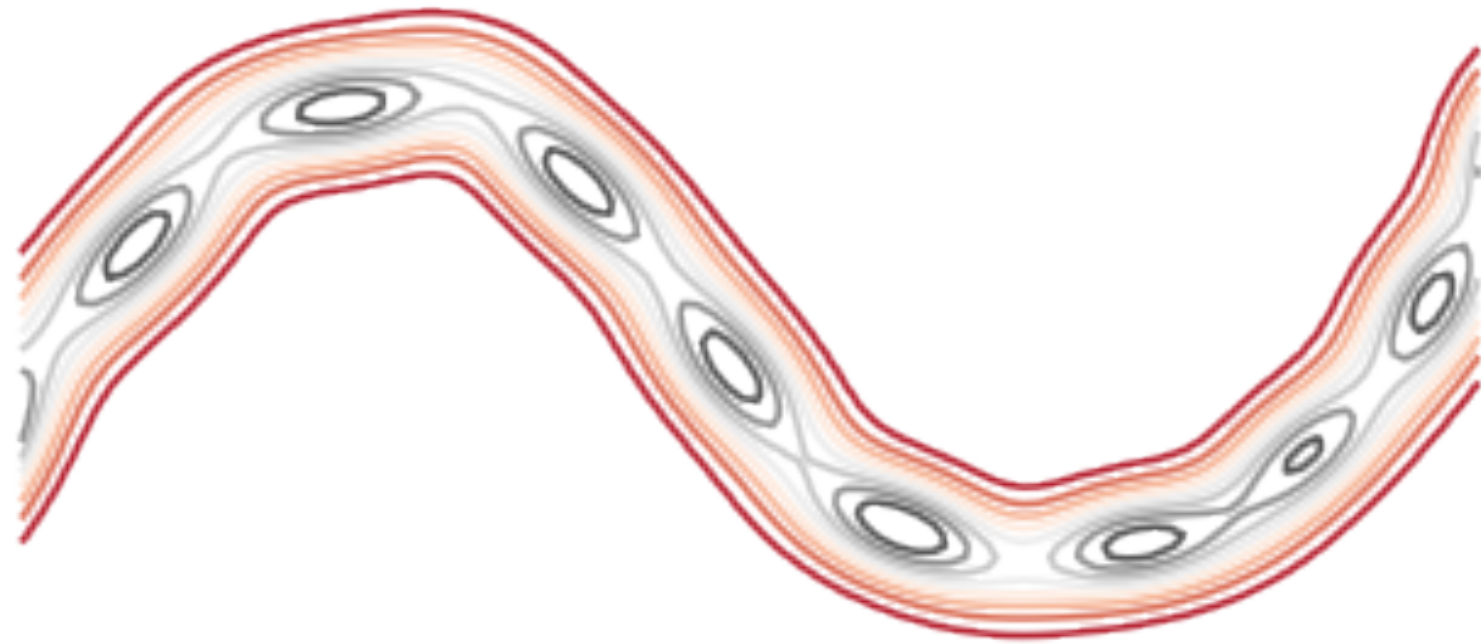
$$Y = AX + N$$

Bayesian framework

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ prior

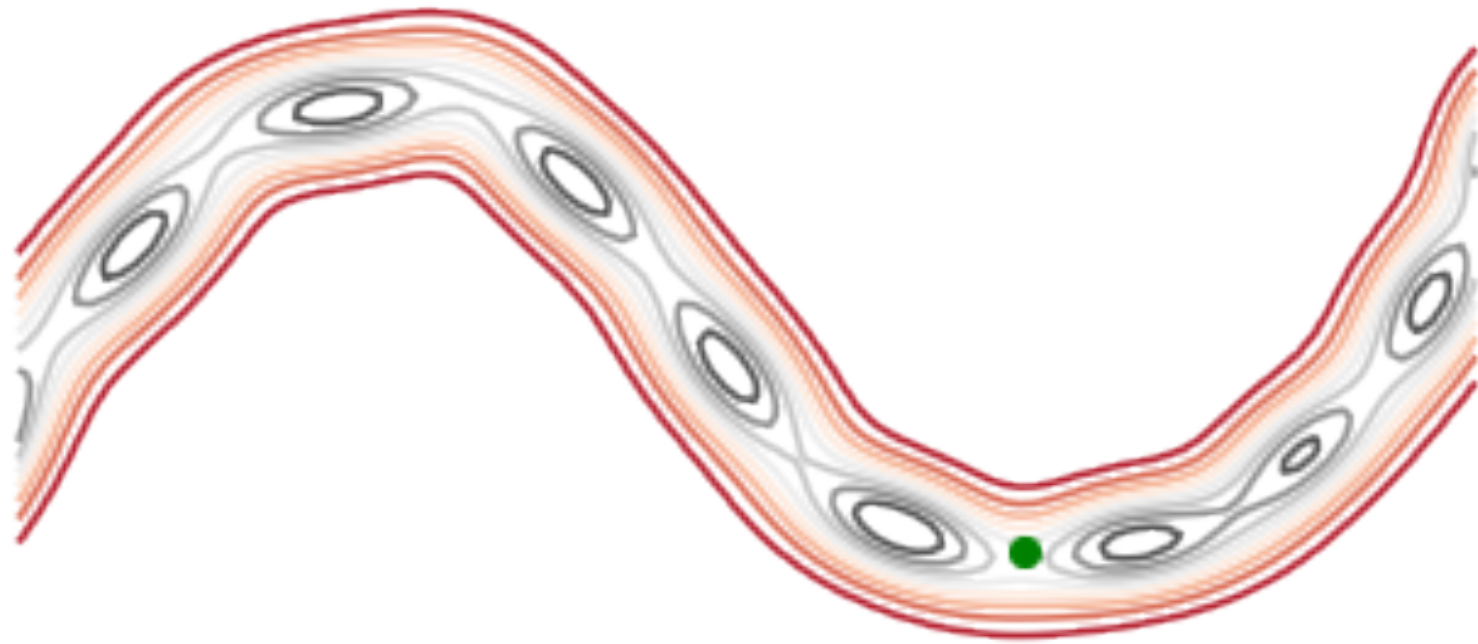


Bayesian framework

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ prior

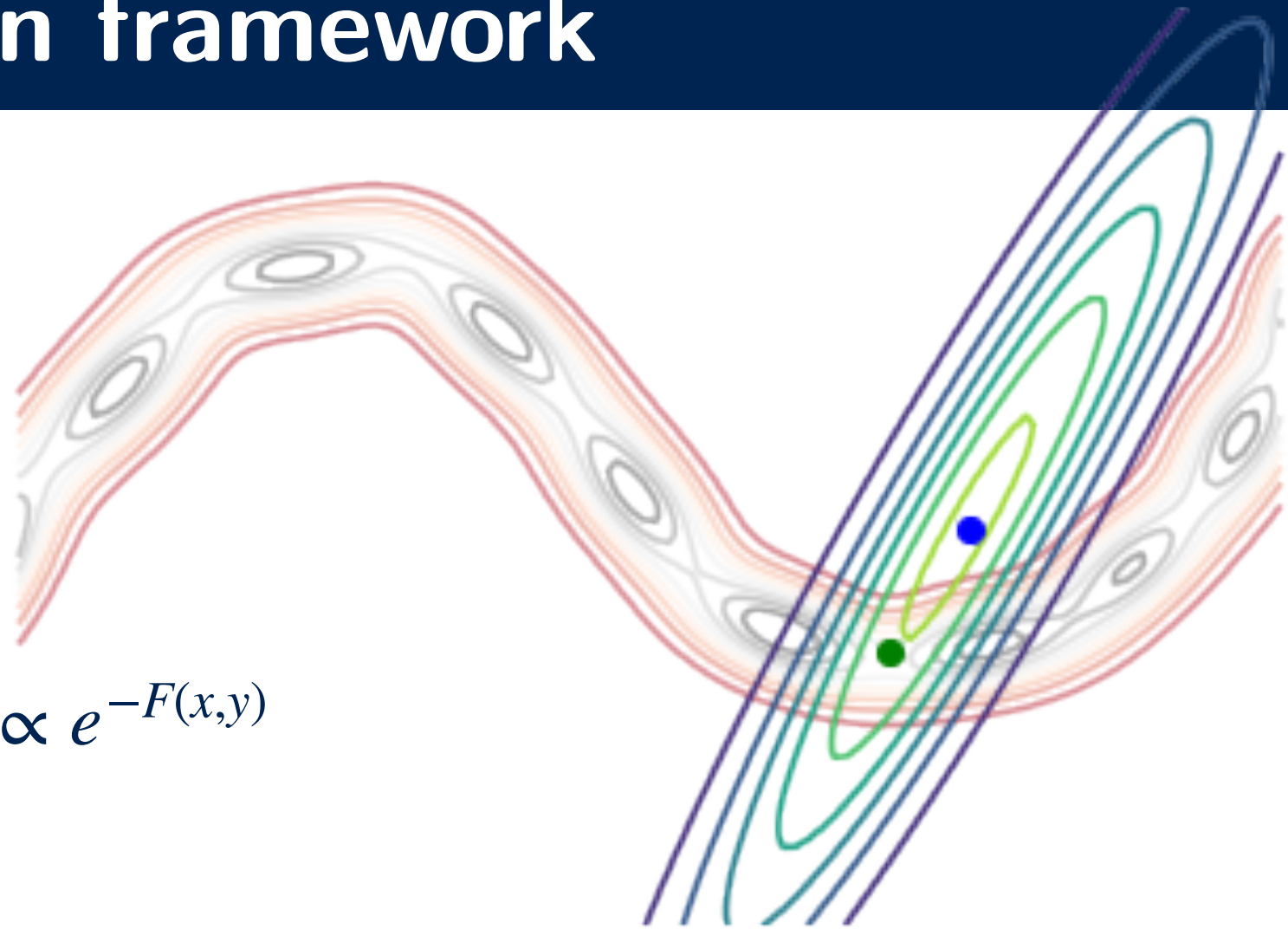


Bayesian framework

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- $Y|X \sim p_{Y|X=x}(y) = p_N(y - Ax) \propto e^{-F(x,y)}$
likelihood



Bayesian framework

Forward degradation model

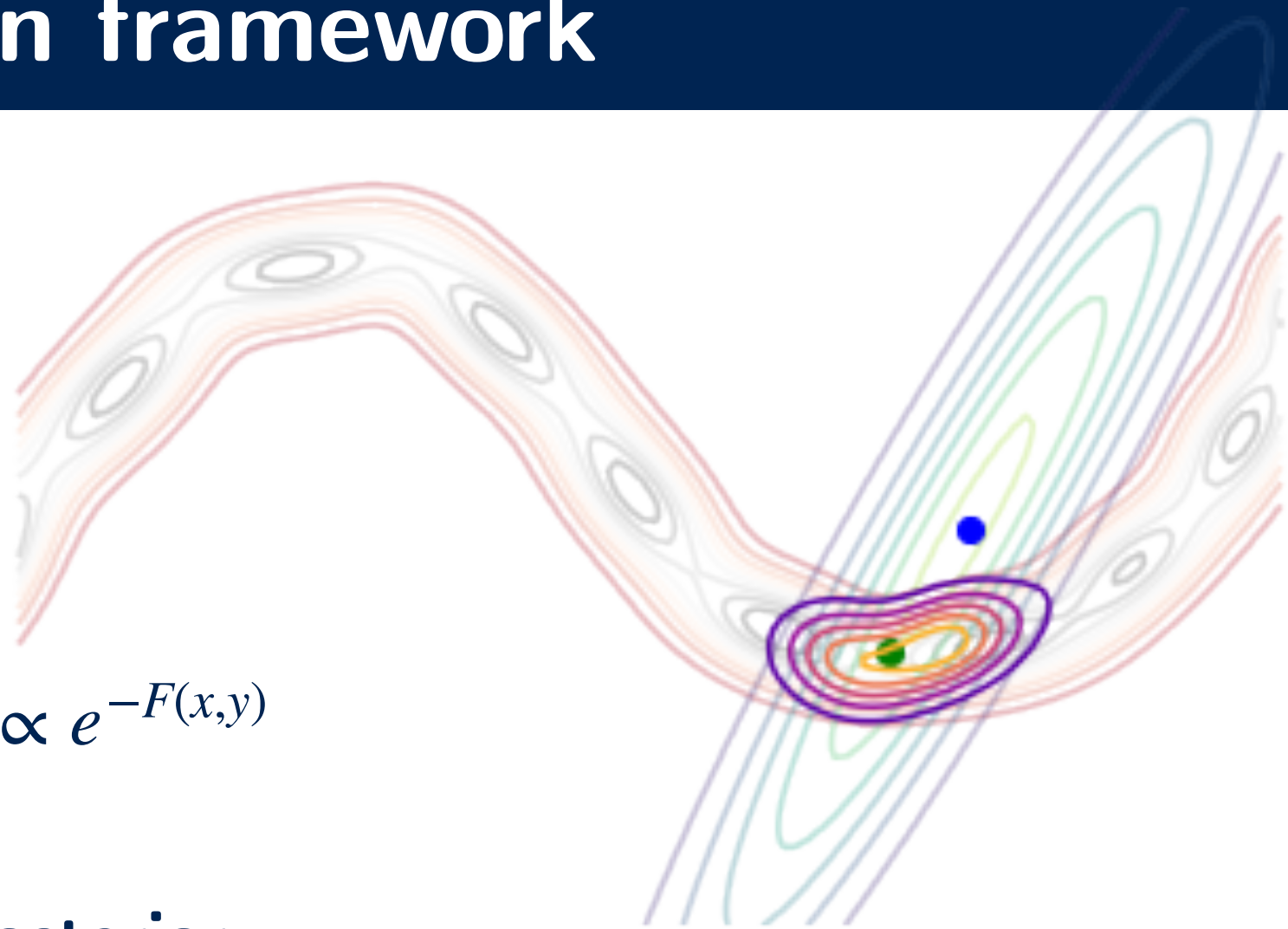
$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- $Y|X \sim p_{Y|X=x}(y) = p_N(y - Ax) \propto e^{-F(x,y)}$

likelihood

- Inferences on x based on the **posterior**

$$p_{X|Y}(x, y) = \frac{p_{Y|X=x}(y)p_X(x)}{p_Y(y)} \propto p_{Y|X=x}(y)p_X(x) \propto e^{-F(x,y)-U(x)}$$



Bayesian framework

Forward degradation model

$$Y = AX + N$$

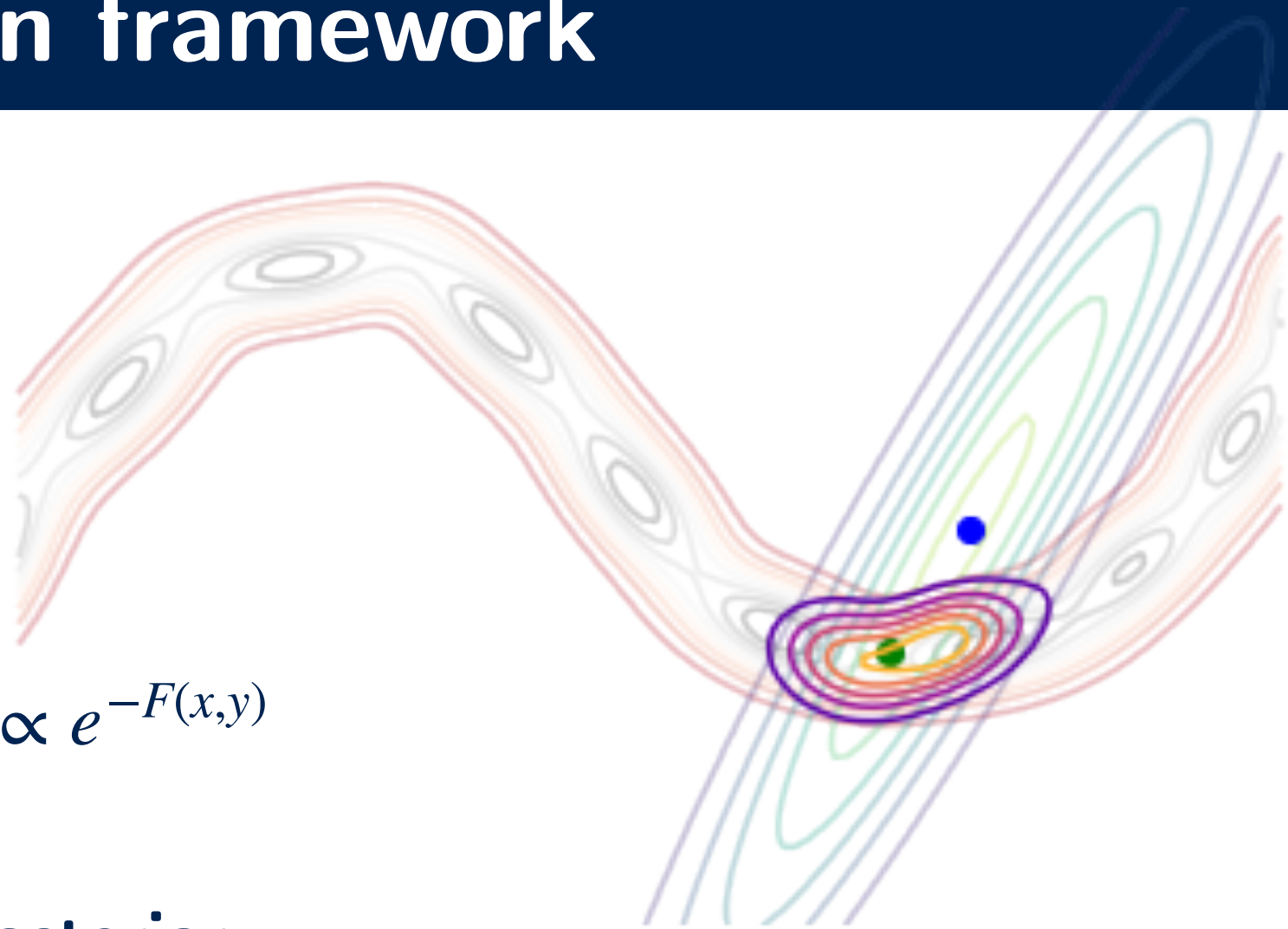
- $X \sim p_X \propto e^{-U}$ **prior**
- $Y|X \sim p_{Y|X=x}(y) = p_N(y - Ax) \propto e^{-F(x,y)}$

likelihood

- Inferences on x based on the **posterior**

$$p_{X|Y}(x, y) = \frac{p_{Y|X=x}(y)p_X(x)}{p_Y(y)} \propto p_{Y|X=x}(y)p_X(x) \propto e^{-F(x,y)-U(x)}$$

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



Bayesian framework

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- $Y|X \sim p_{Y|X=x}(y) = p_N(y - Ax) \propto e^{-F(x,y)}$

likelihood

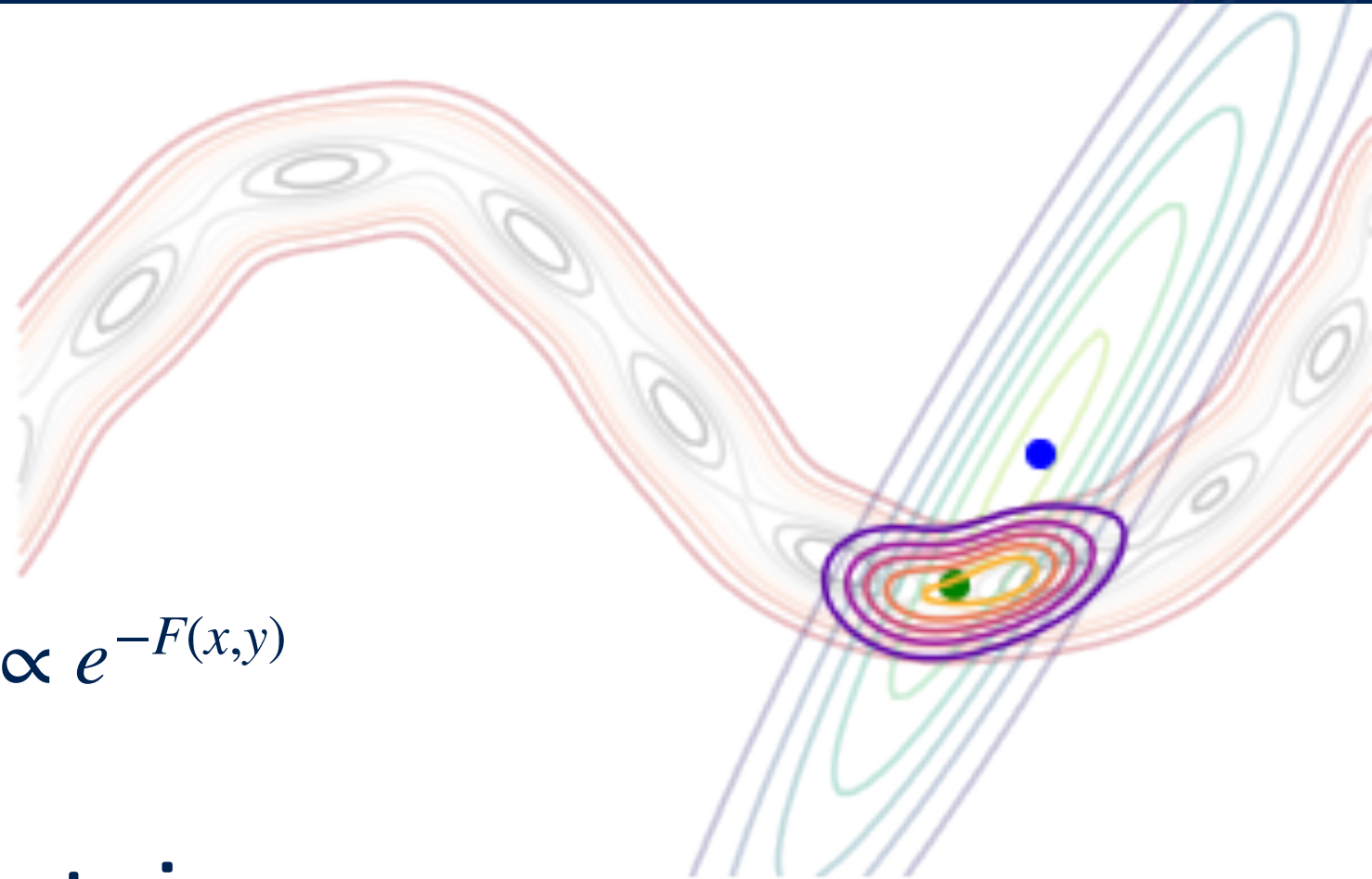
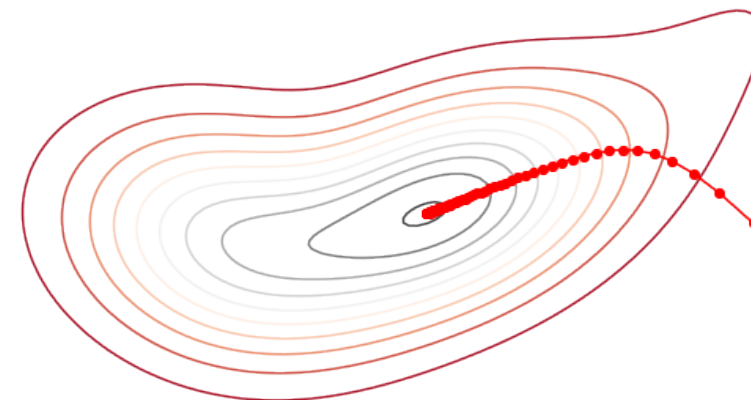
- Inferences on x based on the **posterior**

$$p_{X|Y}(x, y) = \frac{p_{Y|X=x}(y)p_X(x)}{p_Y(y)} \propto p_{Y|X=x}(y)p_X(x) \propto e^{-F(x,y)-U(x)}$$

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

MAP (optimization)

$$\begin{aligned}\hat{x}_{MAP} &= \operatorname{argmax}_x p_{X|Y}(x|y) \\ &= \operatorname{argmin}_{x \in \mathbb{R}^d} F(x, y) + U(x)\end{aligned}$$



Bayesian framework

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- $Y|X \sim p_{Y|X=x}(y) = p_N(y - Ax) \propto e^{-F(x,y)}$

likelihood

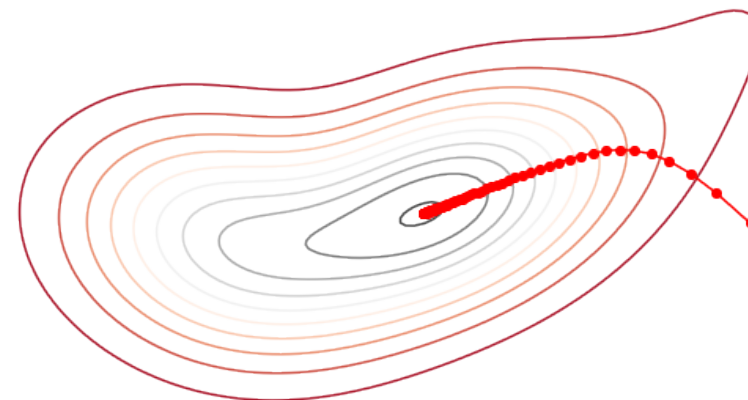
- Inferences on x based on the **posterior**

$$p_{X|Y}(x, y) = \frac{p_{Y|X=x}(y)p_X(x)}{p_Y(y)} \propto p_{Y|X=x}(y)p_X(x) \propto e^{-F(x,y)-U(x)}$$

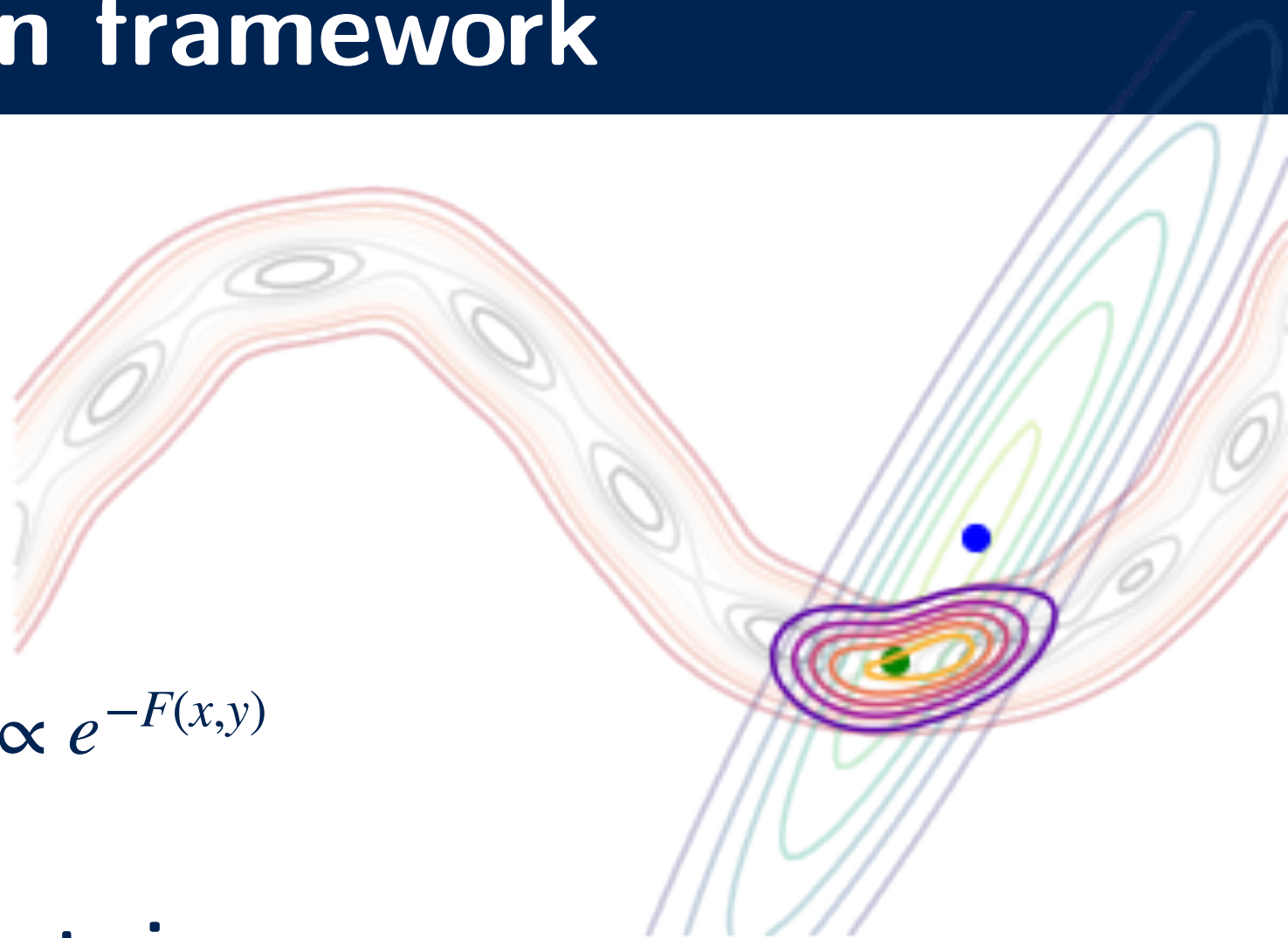
→ models our belief about X after observing $Y = y$.

MAP (optimization)

$$\begin{aligned}\hat{x}_{MAP} &= \operatorname{argmax}_x p_{X|Y}(x|y) \\ &= \operatorname{argmin}_{x \in \mathbb{R}^d} F(x, y) + U(x)\end{aligned}$$



Numerous schemes with **known convergence properties in the convex case.**

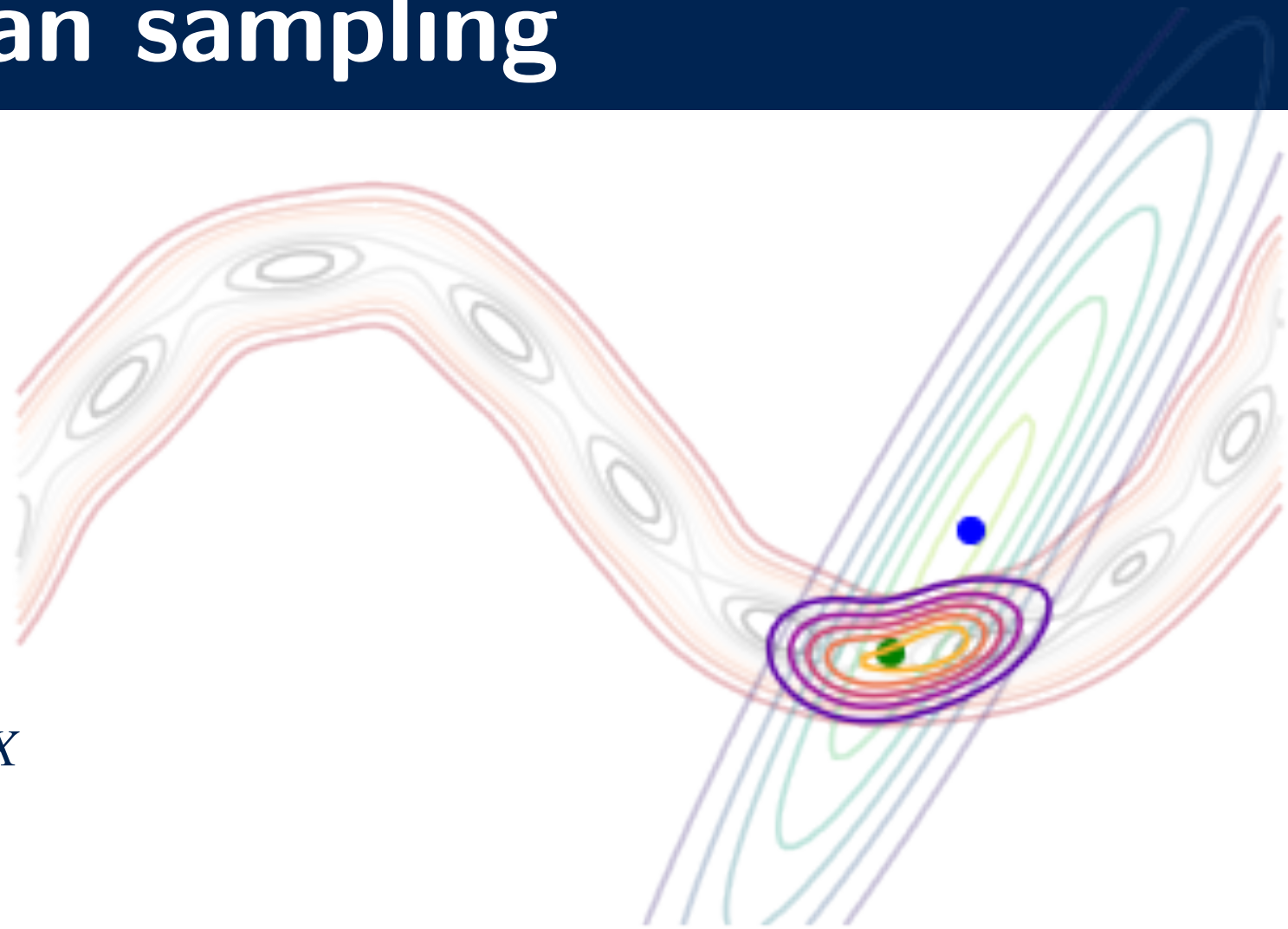


Bayesian sampling

Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X | Y \sim p_{X|Y} \propto p_{Y|X} p_X$



Bayesian sampling

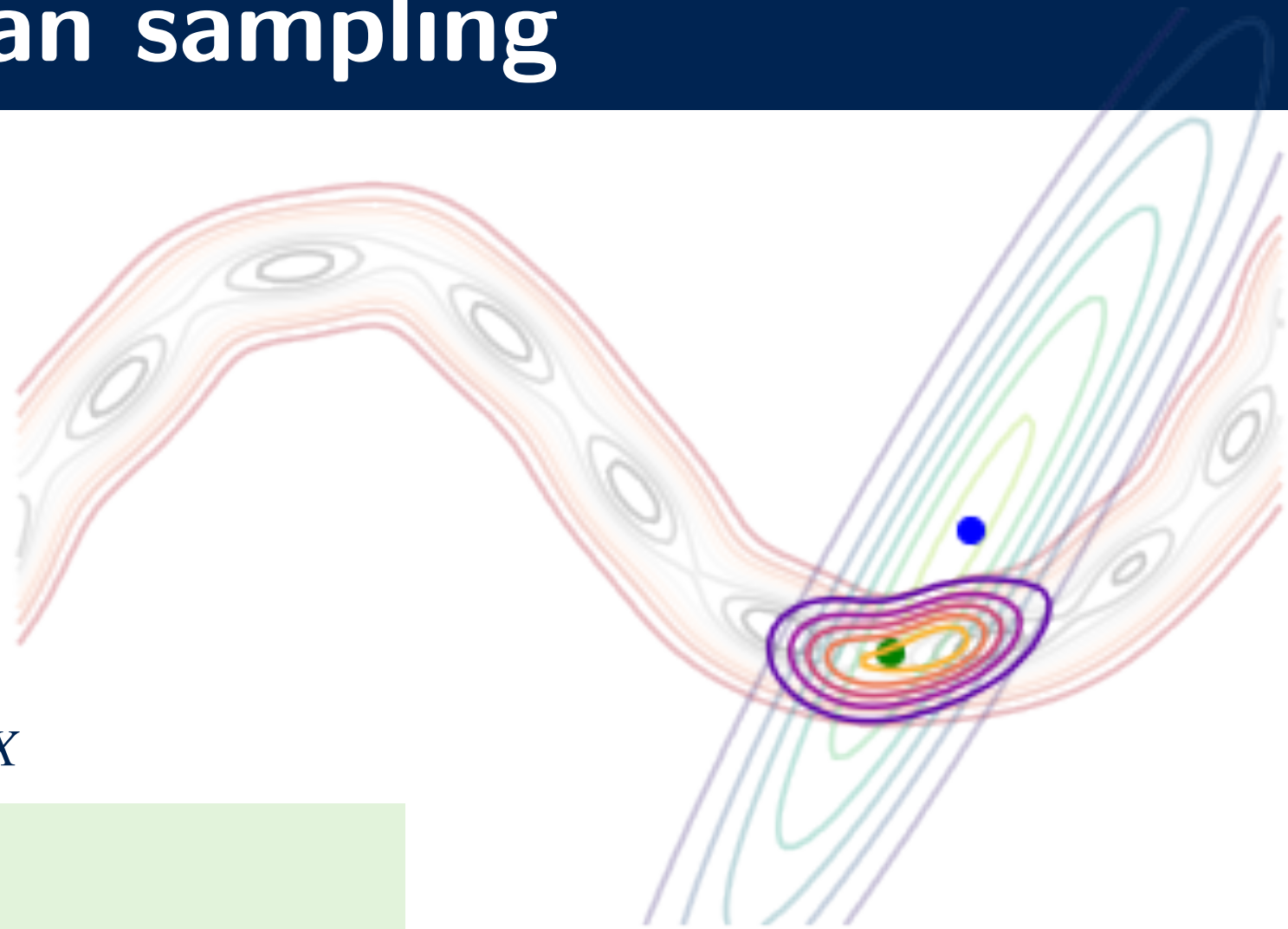
Forward degradation model

$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X|Y \sim p_{X|Y} \propto p_{Y|X}p_X$

MMSE

$$\hat{x}_{MMSE} = \mathbb{E}_X[X|Y = y] = \int \tilde{x} p_{X|Y}(\tilde{x}|y) d\tilde{x}$$



Bayesian sampling

Forward degradation model

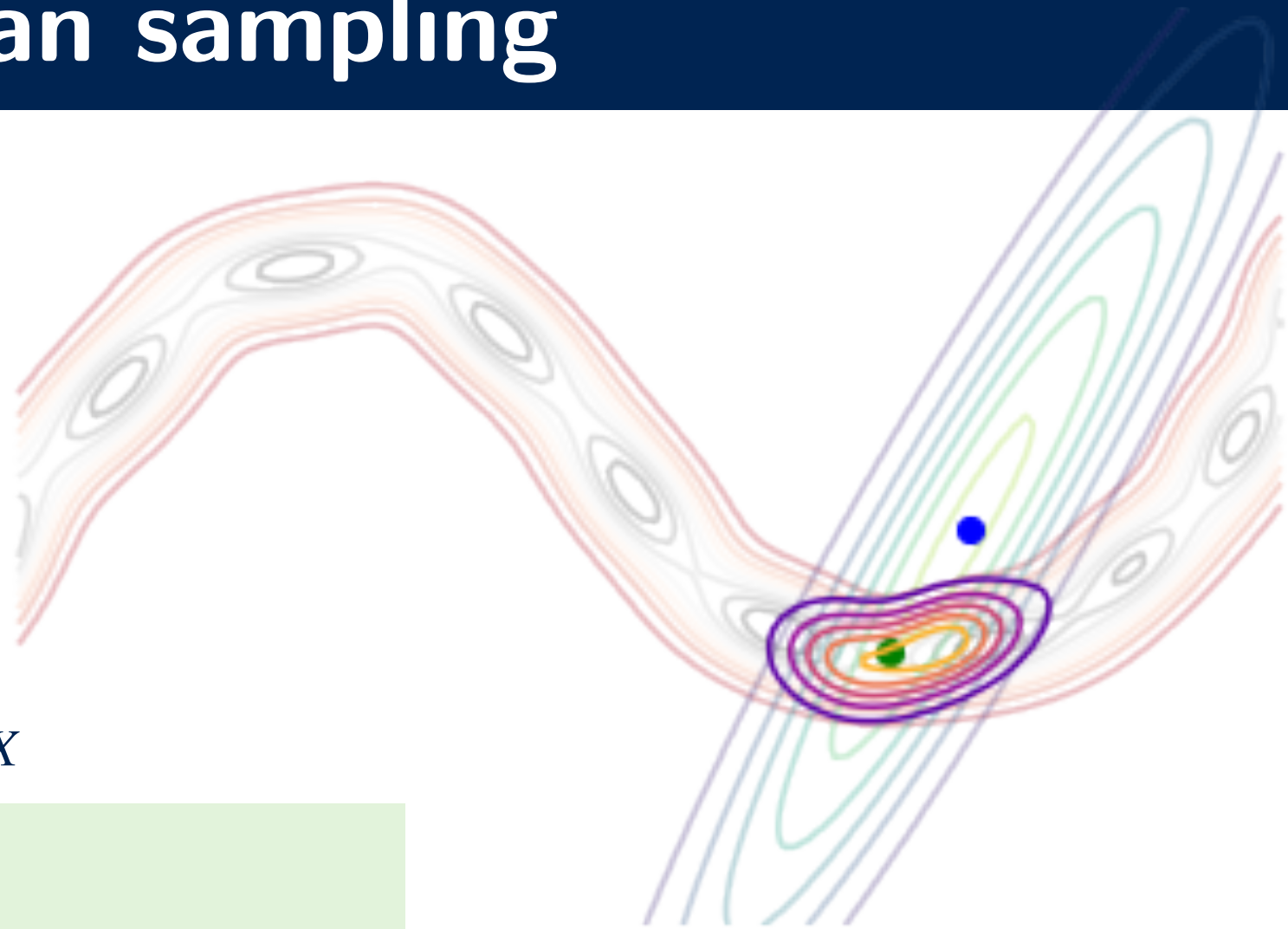
$$Y = AX + N$$

- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X|Y \sim p_{X|Y} \propto p_{Y|X}p_X$

MMSE

$$\hat{x}_{MMSE} = \mathbb{E}_X[X|Y = y] = \int \tilde{x} p_{X|Y}(\tilde{x}|y) d\tilde{x}$$

- Quantify uncertainties
- Estimating $p(X \in A | y)$,
- Hypothesis testing, etc...



Bayesian sampling

Forward degradation model

$$Y = AX + N$$

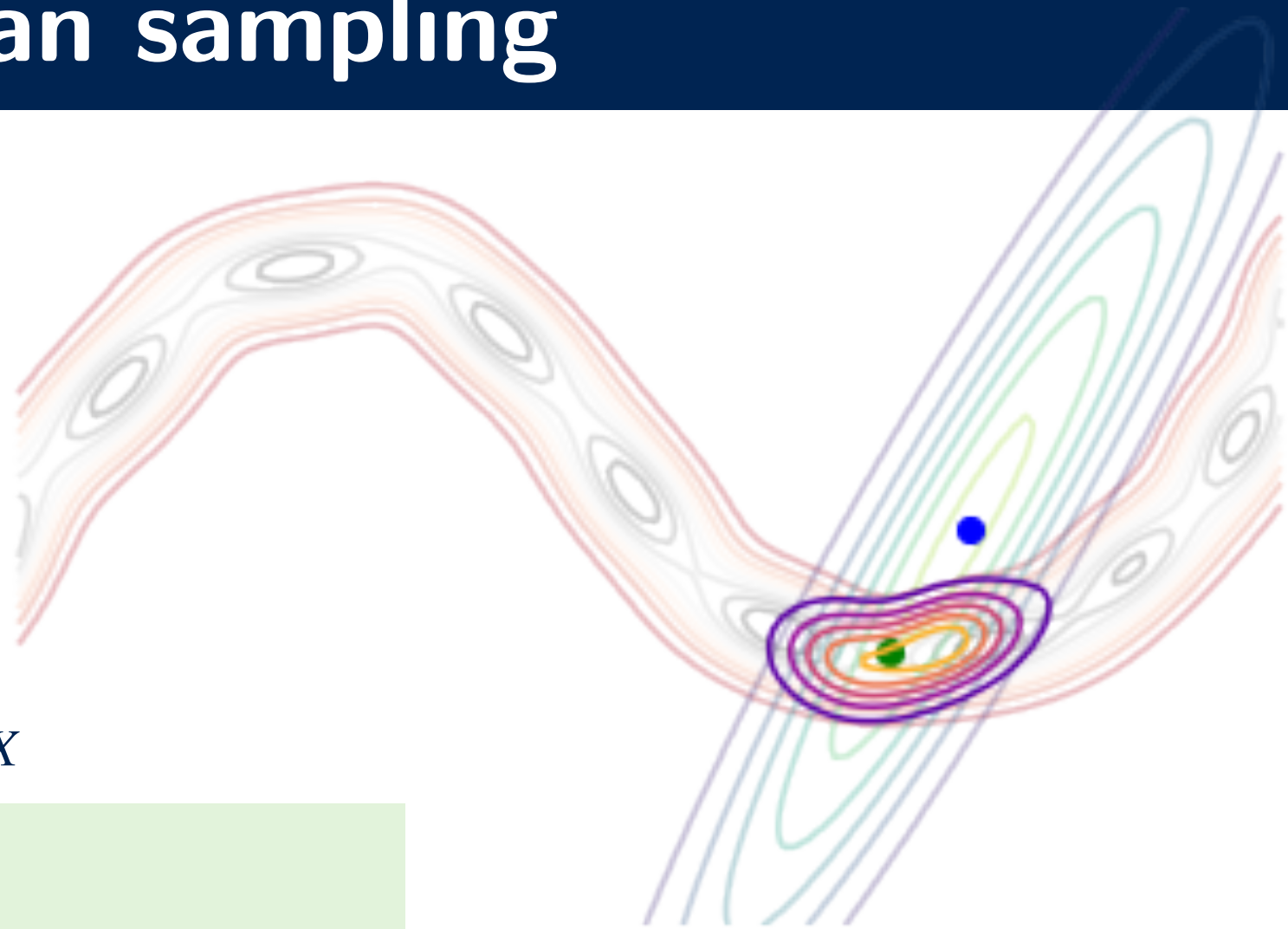
- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X|Y \sim p_{X|Y} \propto p_{Y|X}p_X$

MMSE

$$\hat{x}_{MMSE} = \mathbb{E}_X[X|Y = y] = \int \tilde{x} p_{X|Y}(\tilde{x}|y) d\tilde{x}$$

- Quantify uncertainties
- Estimating $p(X \in A | y)$,
- Hypothesis testing, etc...

➔ **sampling!**



Bayesian sampling

Forward degradation model

$$Y = AX + N$$

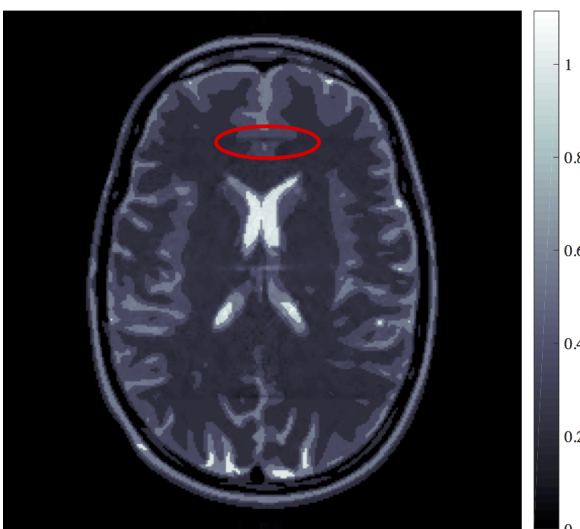
- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X|Y \sim p_{X|Y} \propto p_{Y|X}p_X$

MMSE

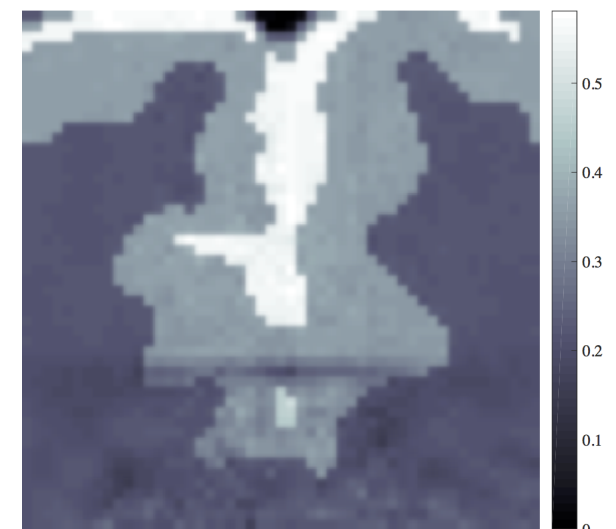
$$\hat{x}_{MMSE} = \mathbb{E}_X[X|Y = y] = \int \tilde{x} p_{X|Y}(\tilde{x}|y) d\tilde{x}$$

- Quantify uncertainties
- Estimating $p(X \in A | y)$,
- Hypothesis testing, etc...

→ **sampling!**

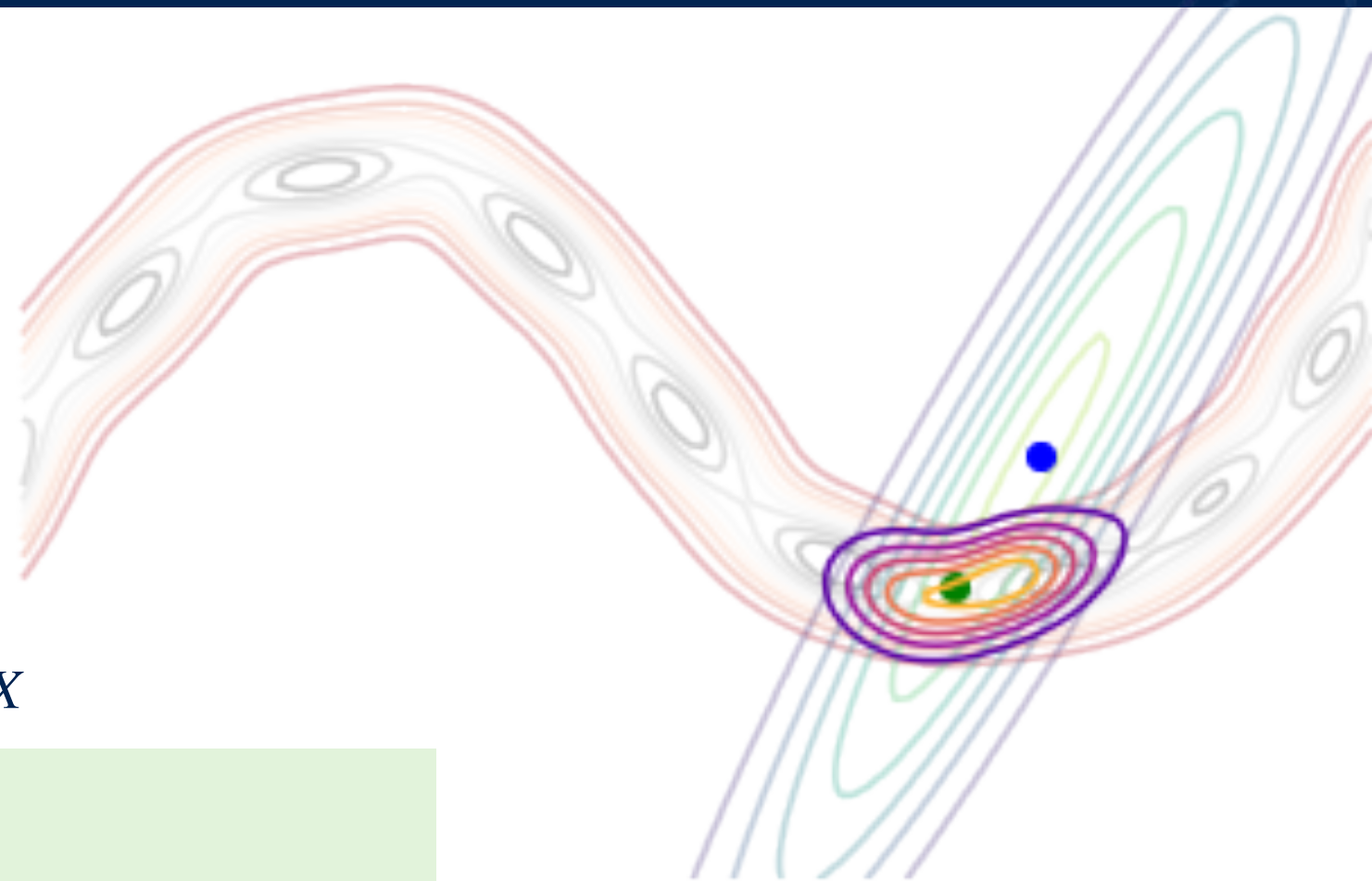


MRI for brain lesion



zoom

[Repetti et al. 2019]



Bayesian sampling

Forward degradation model

$$Y = AX + N$$

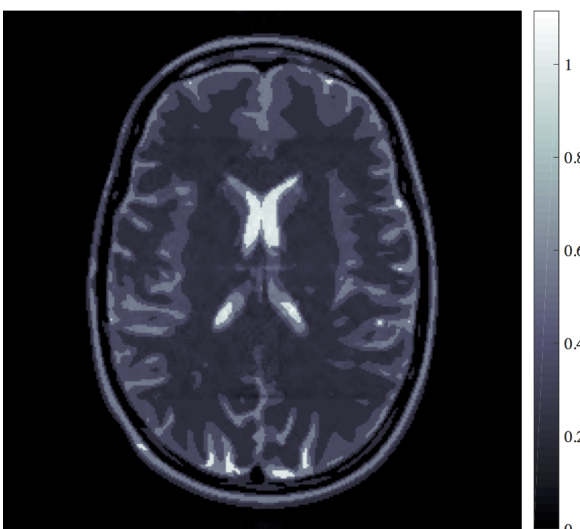
- $X \sim p_X \propto e^{-U}$ **prior**
- **Posterior** $X|Y \sim p_{X|Y} \propto p_{Y|X}p_X$

MMSE

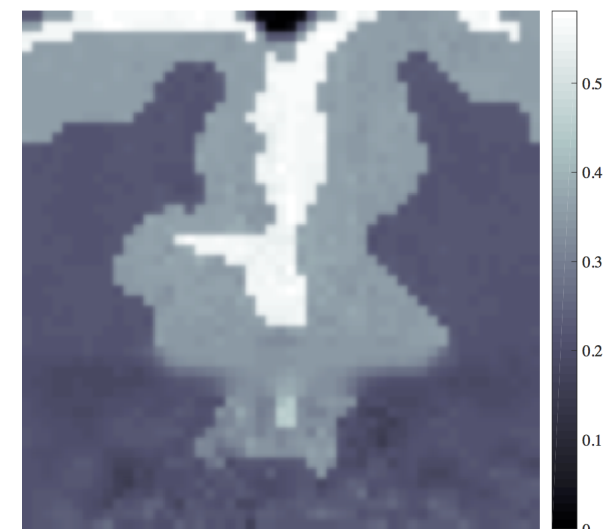
$$\hat{x}_{MMSE} = \mathbb{E}_X[X|Y = y] = \int \tilde{x} p_{X|Y}(\tilde{x}|y) d\tilde{x}$$

- Quantify uncertainties
- Estimating $p(X \in A | y)$,
- Hypothesis testing, etc...

→ **sampling!**

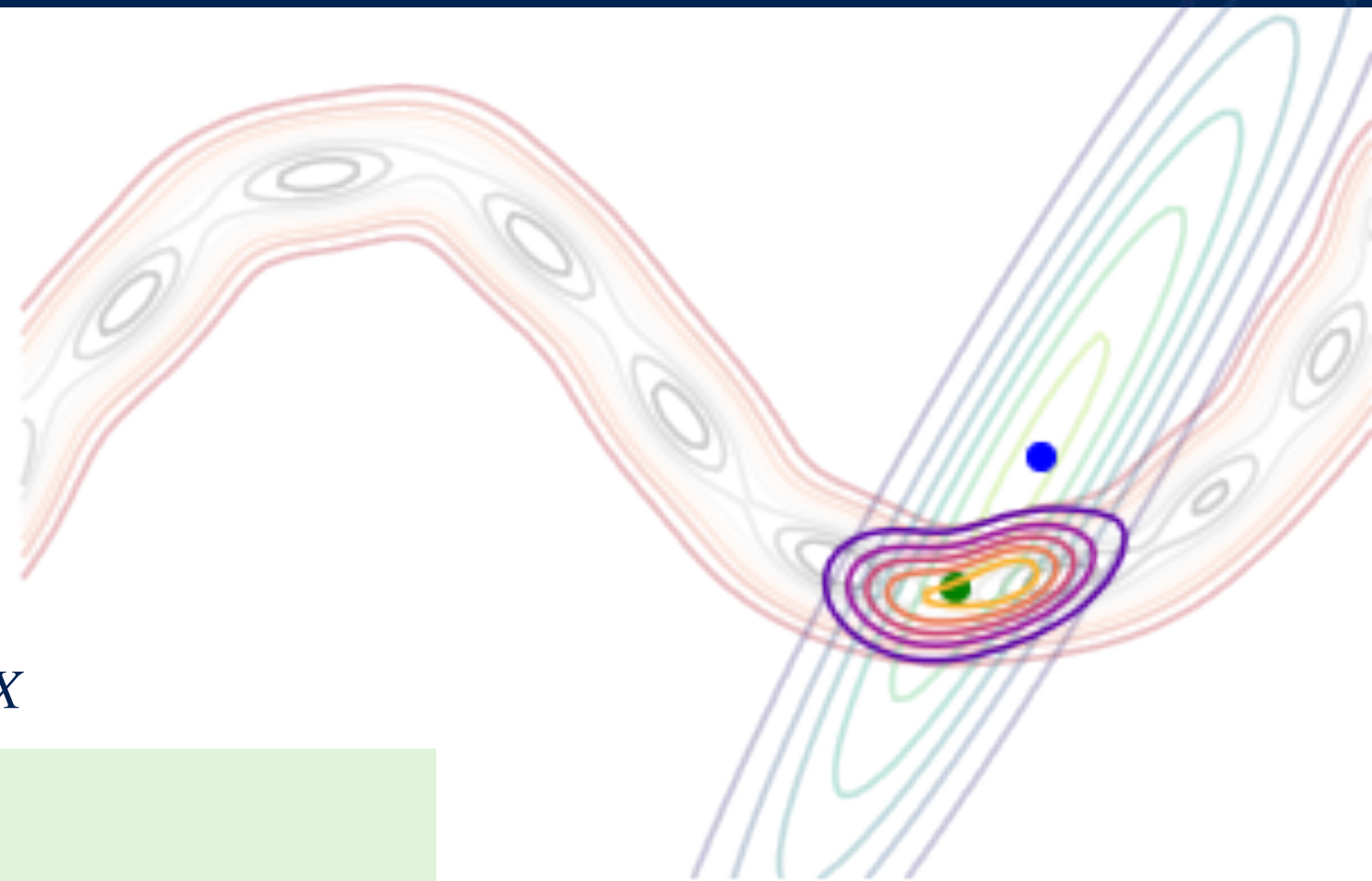


MRI for brain lesion

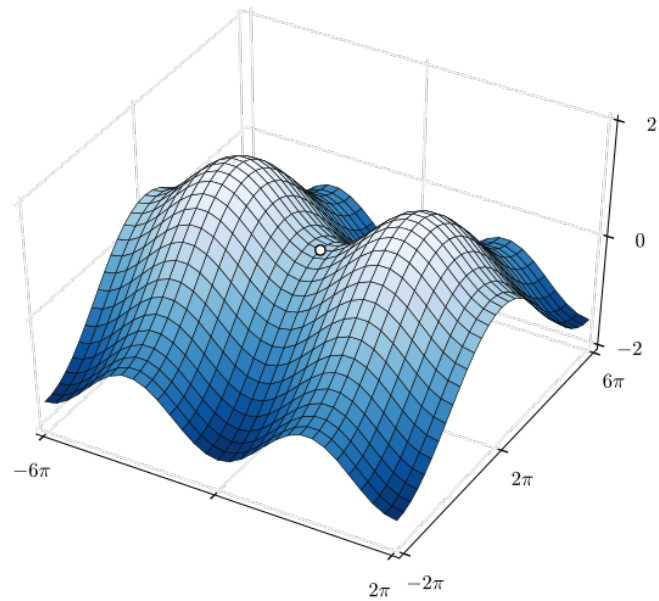


zoom

[Repetti et al. 2019]



Which prior p_X ?



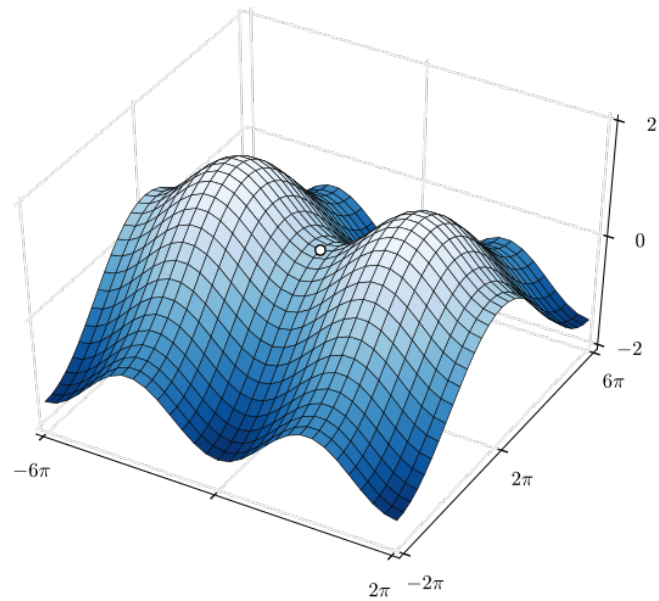
Global models / regularity, $\text{Ex} : U(x) = \alpha \text{TV}(x) = \alpha \sum_i \|\nabla x_i\|$

functional analysis

[Rudin-Osher-Fatemi 1992]



Which prior p_X ?

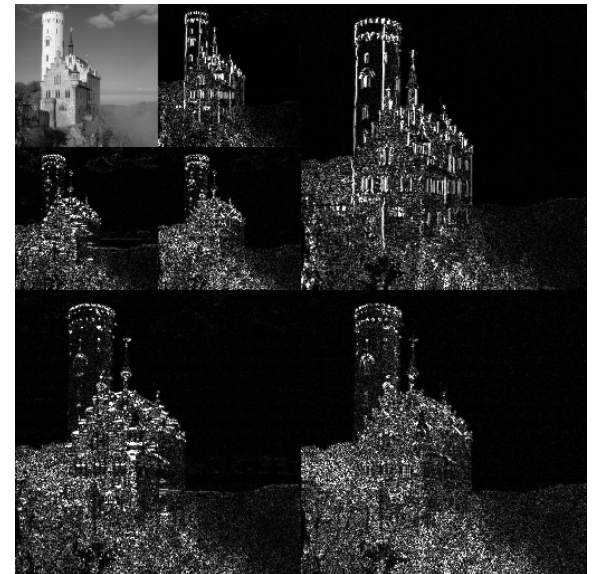
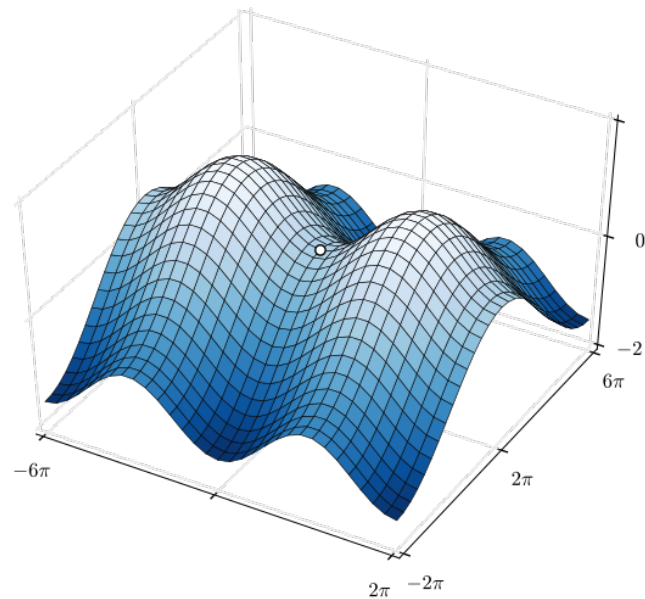


Global models / regularity, Ex : $U(x) = \alpha \text{TV}(x) = \alpha \sum_i \|\nabla x_i\|$
functional analysis

[Rudin-Osher-Fatemi 1992]



Which prior p_X ?



Sparsity in a
transformed
domain

Ex : $U(x) = \|Wx\|_1$

[Donoho-Johnstone 1994]

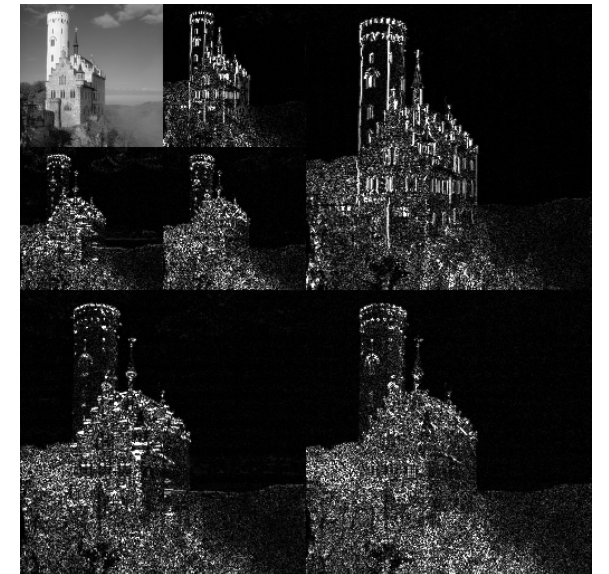
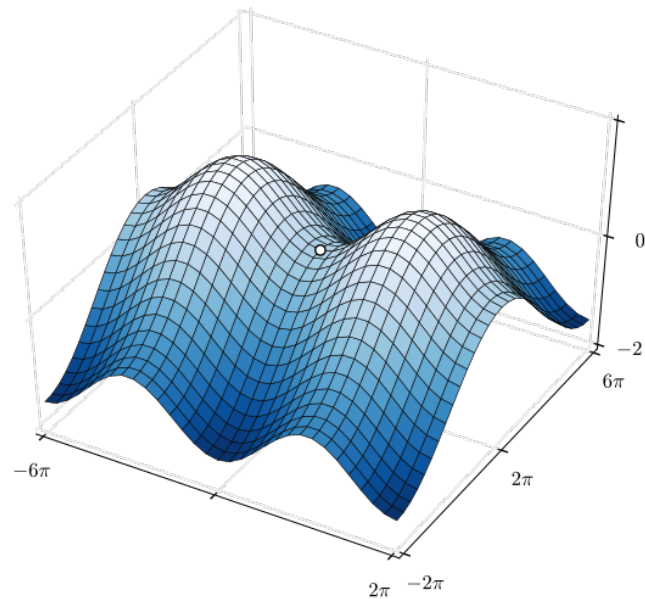
Global models / regularity, Ex : $U(x) = \alpha \text{TV}(x) = \alpha \sum_i \|\nabla x_i\|$

[Rudin-Osher-Fatemi 1992]

functional analysis



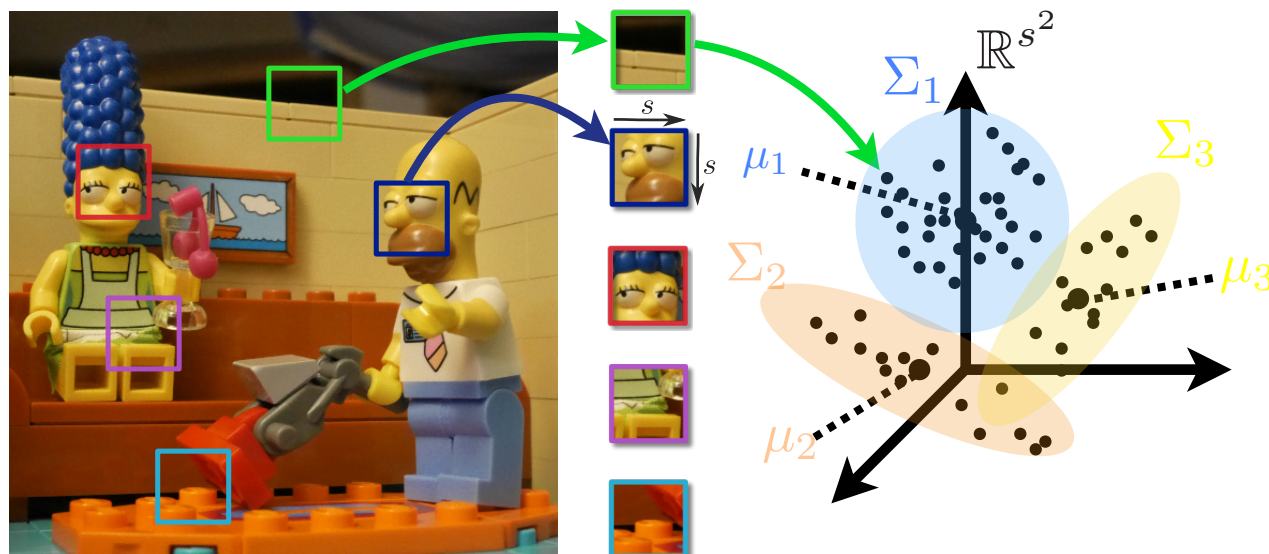
Which prior p_X ?



Sparsity in a transformed domain

Ex : $U(x) = \|Wx\|_1$
[Donoho-Johnstone 1994]

Patch-based models



$$U(x) = \sum_{\text{patches } P_k \text{ of } x} U_k(x(P_k))$$

ex : [Zoran-Weiss 2011],[Houdard et al. 2018], [Teodoro et al 2017]...

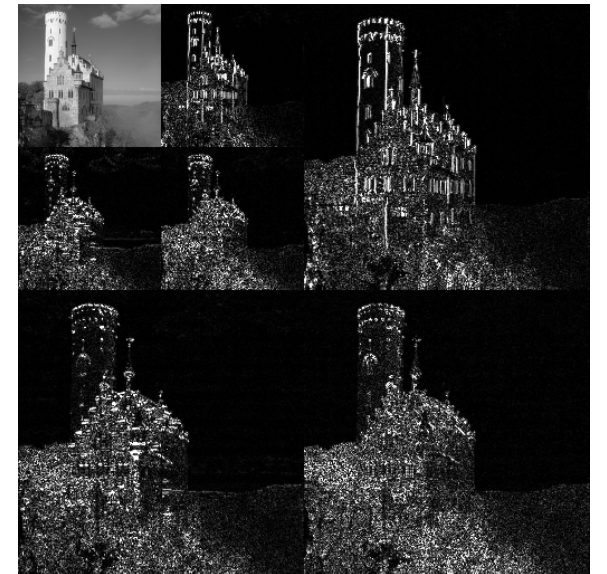
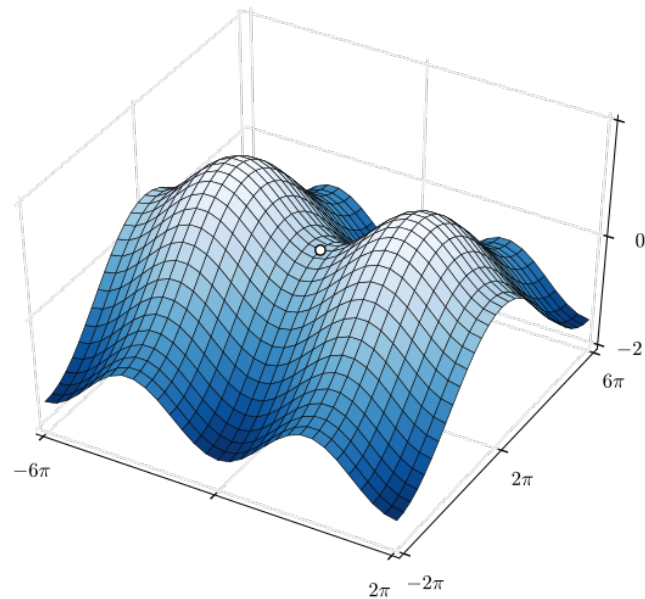
Global models / regularity, $\text{Ex} : U(x) = \alpha \text{TV}(x) = \alpha \sum_i \|\nabla x_i\|$

functional analysis

[Rudin-Osher-Fatemi 1992]



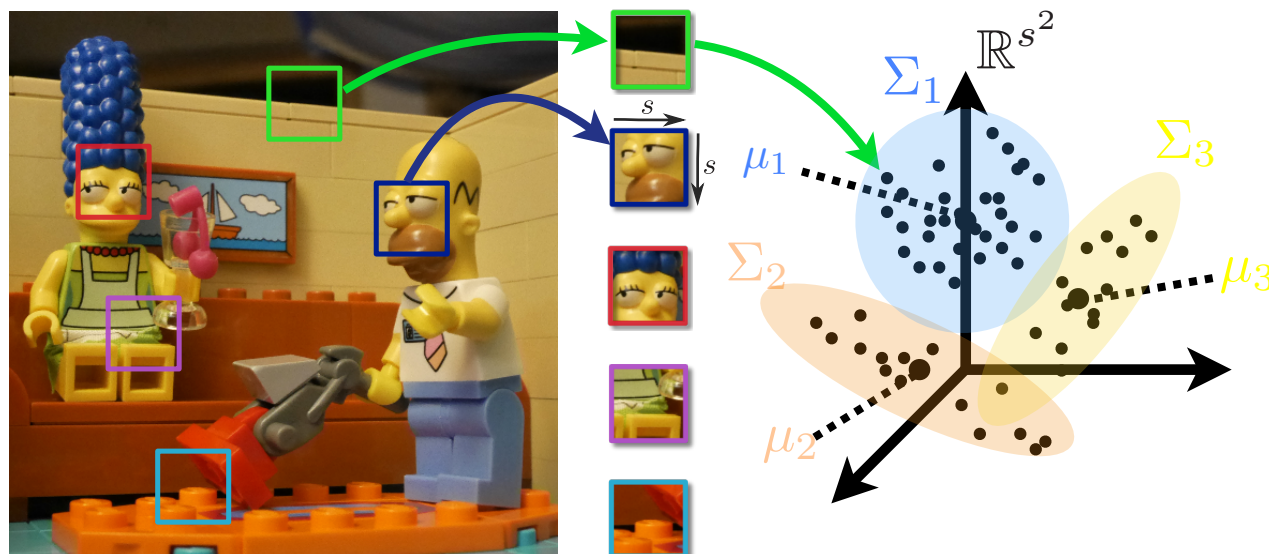
Which prior p_X ?



Sparsity in a transformed domain

$\text{Ex} : U(x) = \|Wx\|_1$
[Donoho-Johnstone 1994]

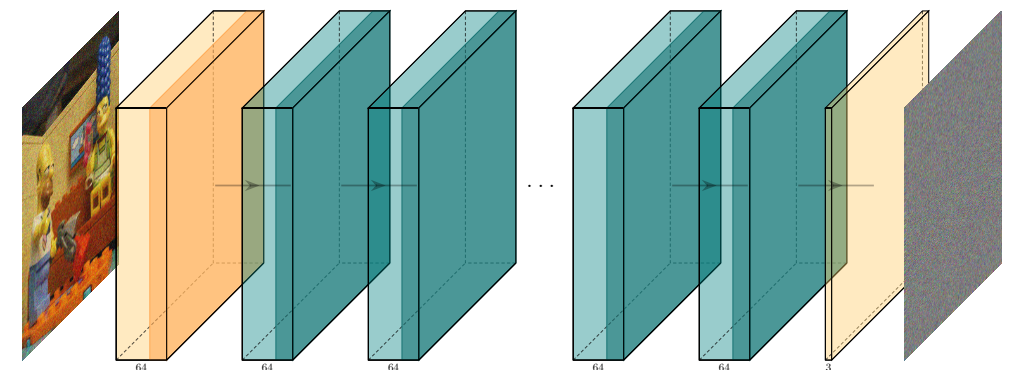
Patch-based models



$$U(x) = \sum_{\text{patches } P_k \text{ of } x} U_k(x(P_k))$$

ex : [Zoran-Weiss 2011], [Houdard et al. 2018], [Teodoro et al 2017]...

Implicit models



Short reminder on convex optimization for inverse problems

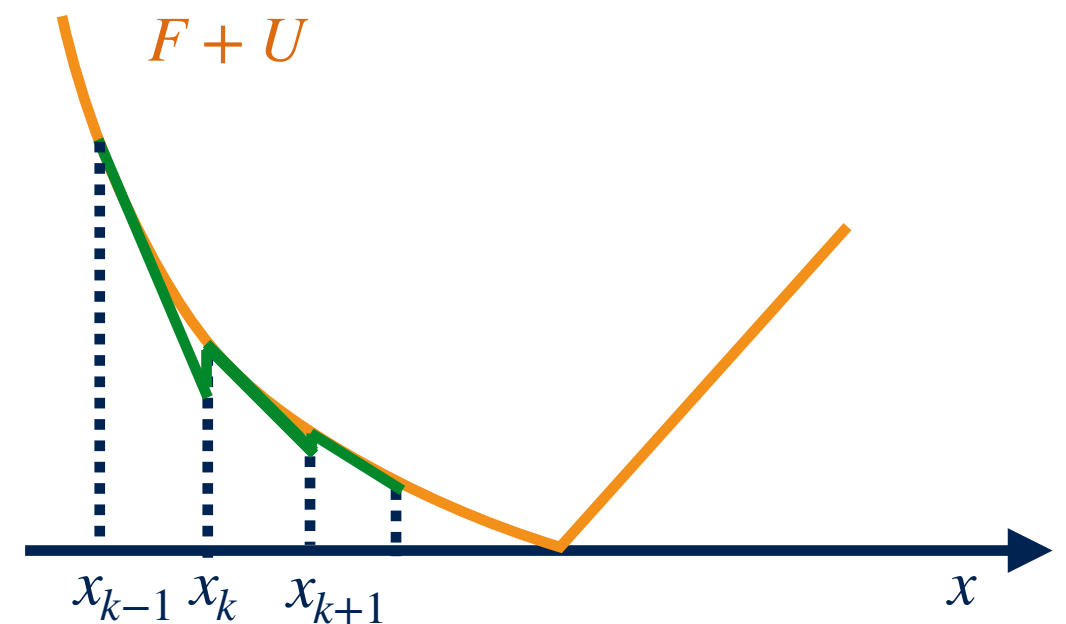
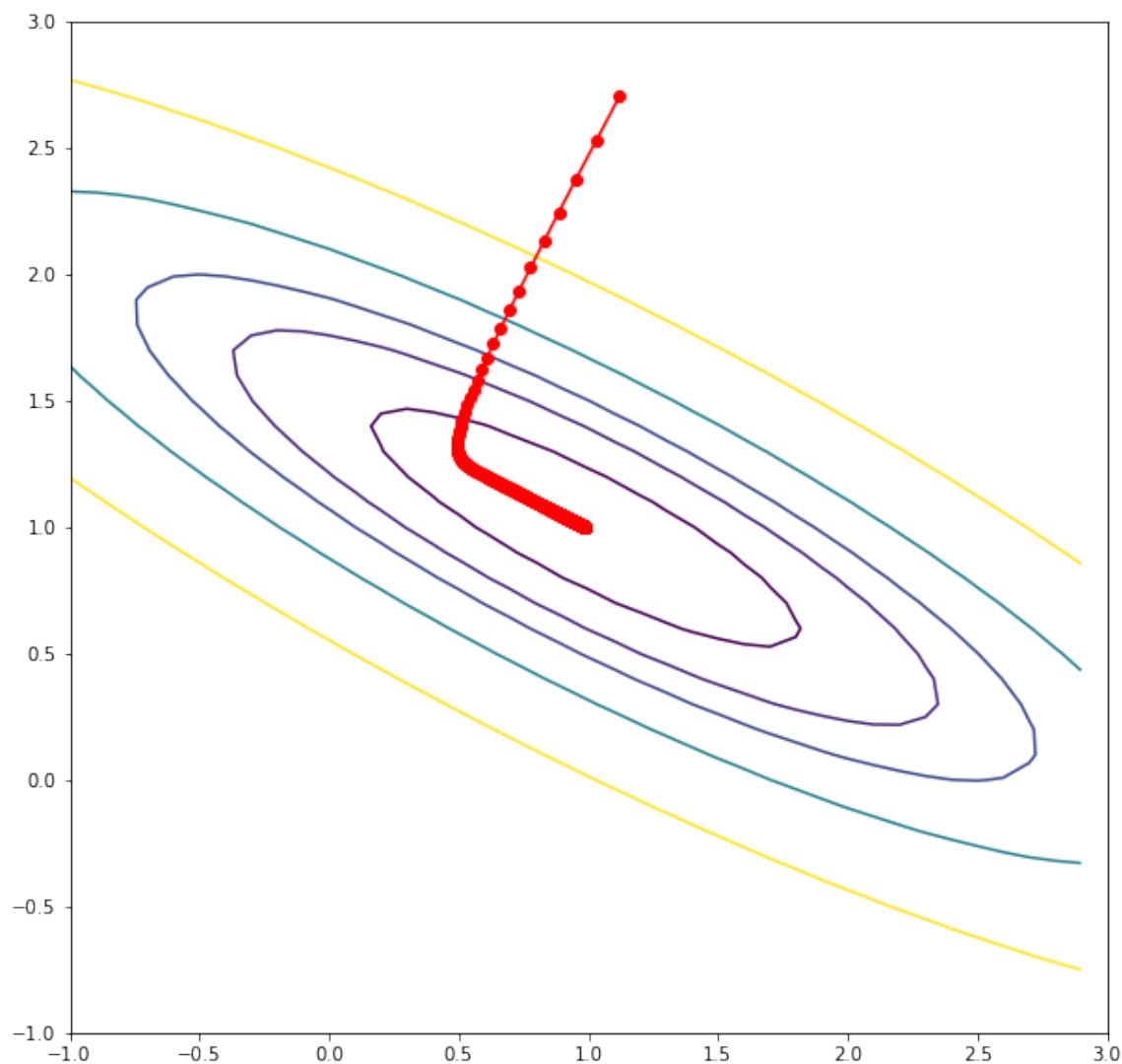
From inverse problems to optimization

$$\arg \min_x \underbrace{\frac{1}{2\sigma^2} \|y - Ax\|_2^2}_{F(x,y)} + U(x)$$

From inverse problems to optimization

$$\arg \min_x \underbrace{\frac{1}{2\sigma^2} \|y - Ax\|_2^2}_{F(x,y)} + U(x)$$

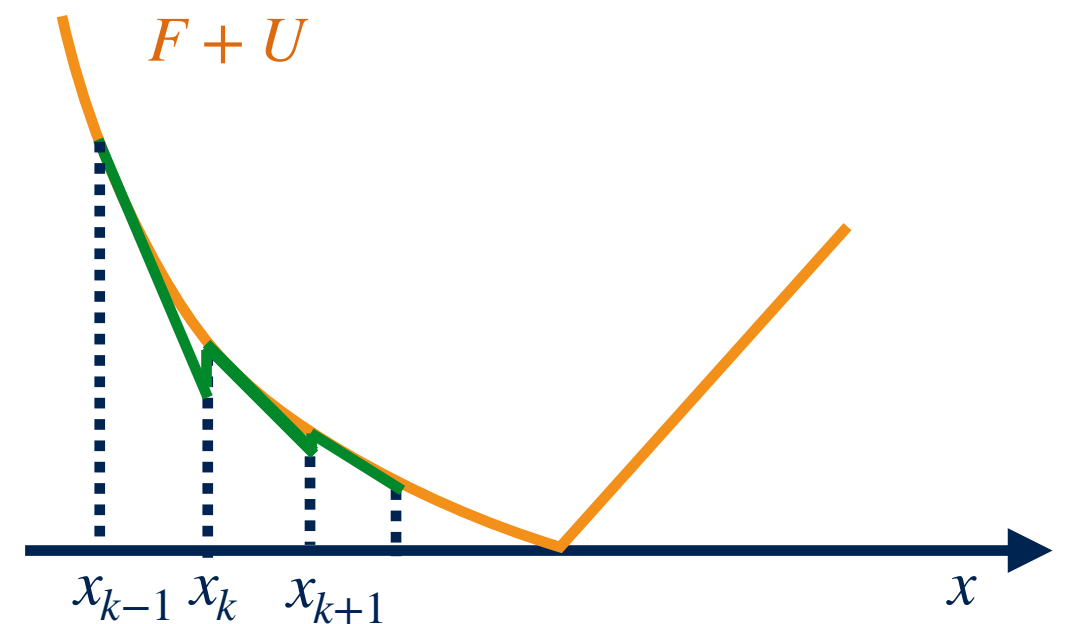
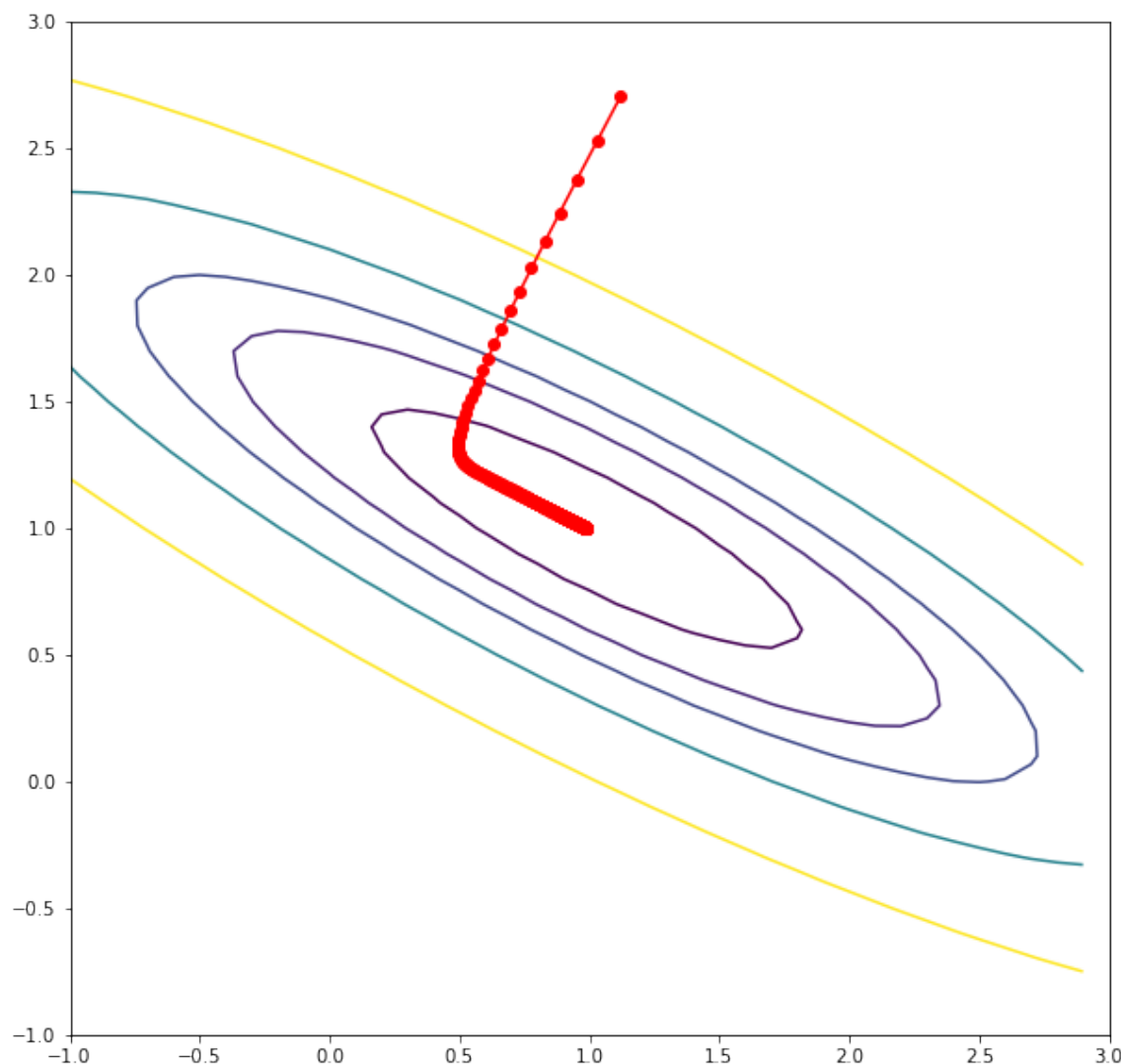
Gradient Descent $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$



From inverse problems to optimization

$$\arg \min_x \underbrace{\frac{1}{2\sigma^2} \|y - Ax\|_2^2}_{F(x,y)} + U(x)$$

Gradient Descent $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$



What if U is not smooth?

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_k) \quad (\text{explicit GD}) \quad \text{for } f \text{ smooth}$$

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_{k+1}) \text{ (implicit GD)} \quad \text{for } f \text{ smooth}$$

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$x_{k+1} = x_k - \gamma \nabla f(x_{k+1})$ (implicit GD) for f smooth

$$\Leftrightarrow x_{k+1} = \arg \min_t \frac{1}{2} \|t - x_k\|_2^2 + \gamma f(t)$$

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_{k+1}) \text{ (implicit GD)} \quad \text{for } f \text{ smooth}$$

$$\Leftrightarrow x_{k+1} = \arg \min_t \underbrace{\frac{1}{2} \|t - x_k\|_2^2 + \gamma f(t)}_{\text{prox}_{\gamma f}(x_k)}$$

Defined even for f not smooth!

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_{k+1}) \text{ (implicit GD)} \quad \text{for } f \text{ smooth}$$

$$\Leftrightarrow x_{k+1} = \arg \min_t \underbrace{\frac{1}{2} \|t - x_k\|_2^2 + \gamma f(t)}_{\text{prox}_{\gamma f}(x_k)}$$

Defined even for f not smooth!

Algorithm x_0 initialization

$$x_{k+1} = \text{prox}_{\gamma f}(x_k)$$

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_{k+1}) \text{ (implicit GD) for } f \text{ smooth}$$

$$\Leftrightarrow x_{k+1} = \arg \min_t \underbrace{\frac{1}{2} \|t - x_k\|_2^2 + \gamma f(t)}_{\text{prox}_{\gamma f}(x_k)}$$

Defined even for f not smooth!

Algorithm x_0 initialization

$$x_{k+1} = \text{prox}_{\gamma f}(x_k)$$

Property $\bar{x} \in \arg \min f \Leftrightarrow \bar{x} = \text{prox}_{\gamma f}(\bar{x})$

Non smooth optim. / Proximal operator

$\arg \min_x f(x)$ with f convex

$$x_{k+1} = x_k - \gamma \nabla f(x_{k+1}) \quad (\text{implicit GD}) \quad \text{for } f \text{ smooth}$$

$$\Leftrightarrow x_{k+1} = \arg \min_t \frac{1}{2} \|t - x_k\|_2^2 + \gamma f(t)$$

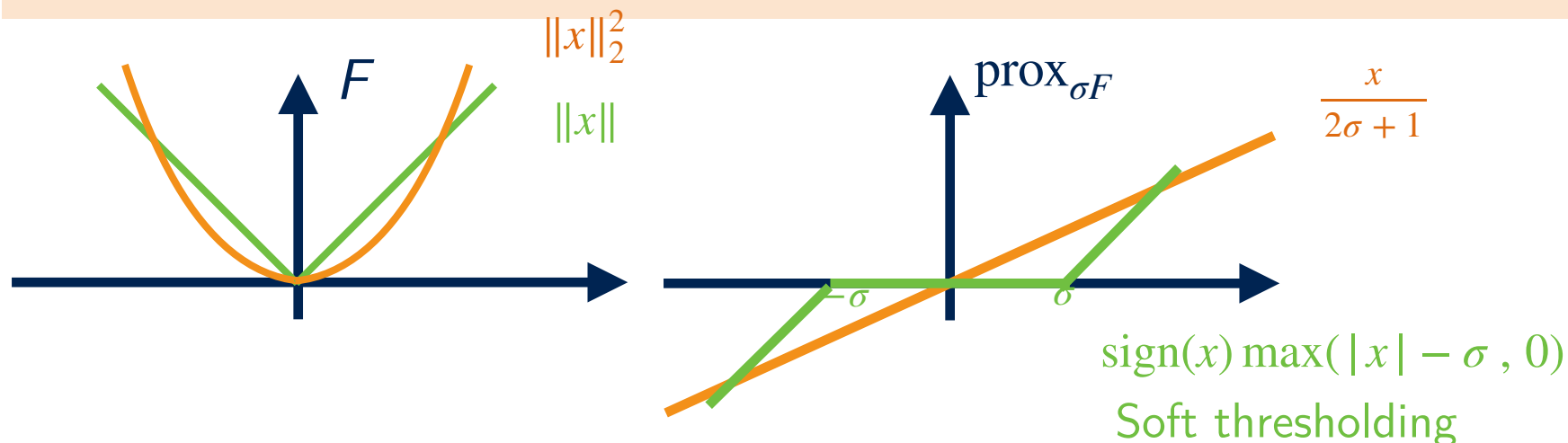
$\text{prox}_{\gamma f}(x_k)$

Defined even for f not smooth!

Algorithm x_0 initialization

$$x_{k+1} = \text{prox}_{\gamma f}(x_k)$$

Property $\bar{x} \in \arg \min f \Leftrightarrow \bar{x} = \text{prox}_{\gamma f}(\bar{x})$



Proximal optimization for inverse pb.

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

Gradient Descent: $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$

Proximal optimization for inverse pb.

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

Gradient Descent: $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$

Proximal gradient (Forward-Backward splitting)

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

Proximal optimization for inverse pb.

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

Gradient Descent: $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$

Proximal gradient (Forward-Backward splitting)

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

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

Proximal optimization for inverse pb.

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

Gradient Descent: $x_{k+1} = x_k - \gamma \nabla F(x_k, y) - \gamma \nabla U(x_k)$

Proximal gradient (Forward-Backward splitting)

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

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

Known convergence properties in the convex case.

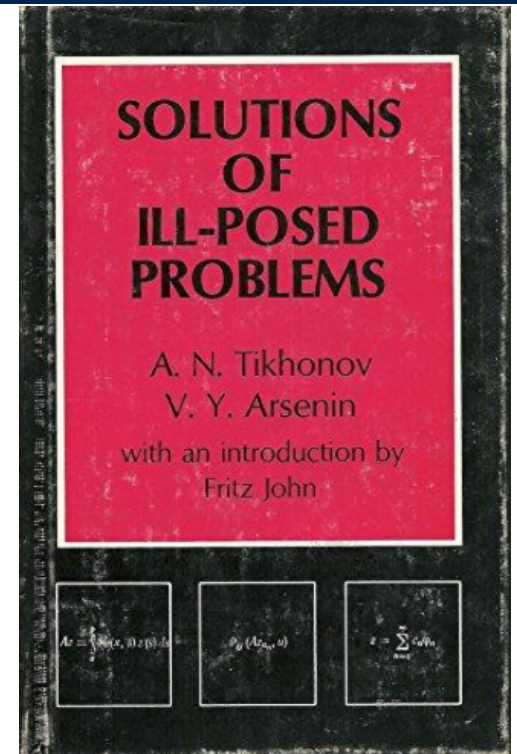
[Moreau 1965], [Combettes-Pesquet 2011], [Chambolle-Pock, 2011]...

Global image models

Tikhonov

[Tikhonov, Arsenin (1977)] $R(x) = \sum_i \|\nabla x_i\|^2$

$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \lambda \sum_i \|\nabla x_i\|_2^2}_{E(x)}$$



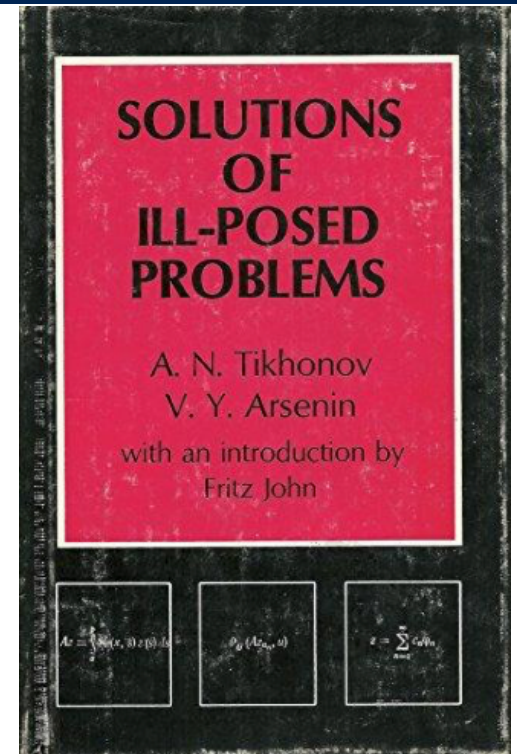
Tikhonov

[Tikhonov, Arsenin (1977)] $R(x) = \sum_i \|\nabla x_i\|^2$

$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \lambda \sum_i \|\nabla x_i\|_2^2}_{E(x)}$$

convex, differentiable, gradient descent with

$$\frac{\partial E}{\partial x} = A^*(Ax - y) - 2\lambda \Delta x$$



Tikhonov

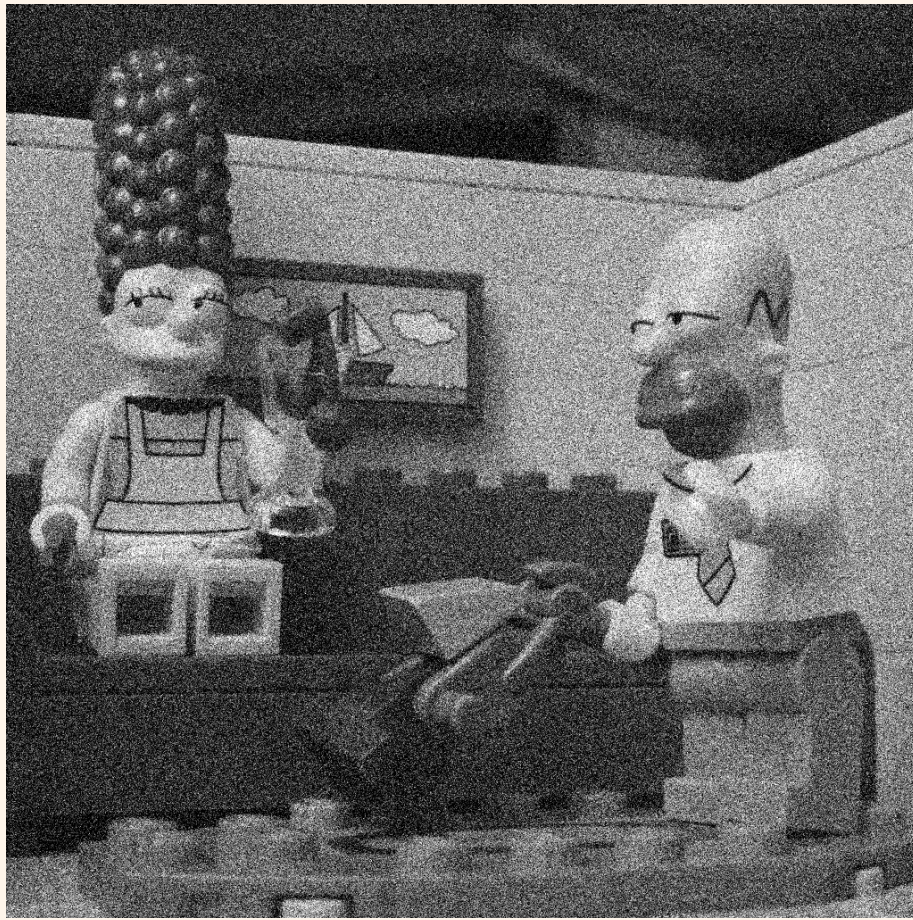
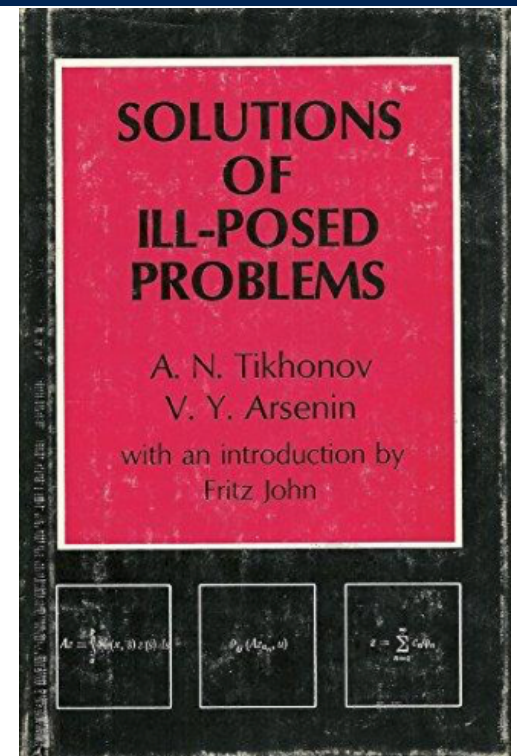
[Tikhonov, Arsenin (1977)]

$$R(x) = \sum_i \|\nabla x_i\|^2$$

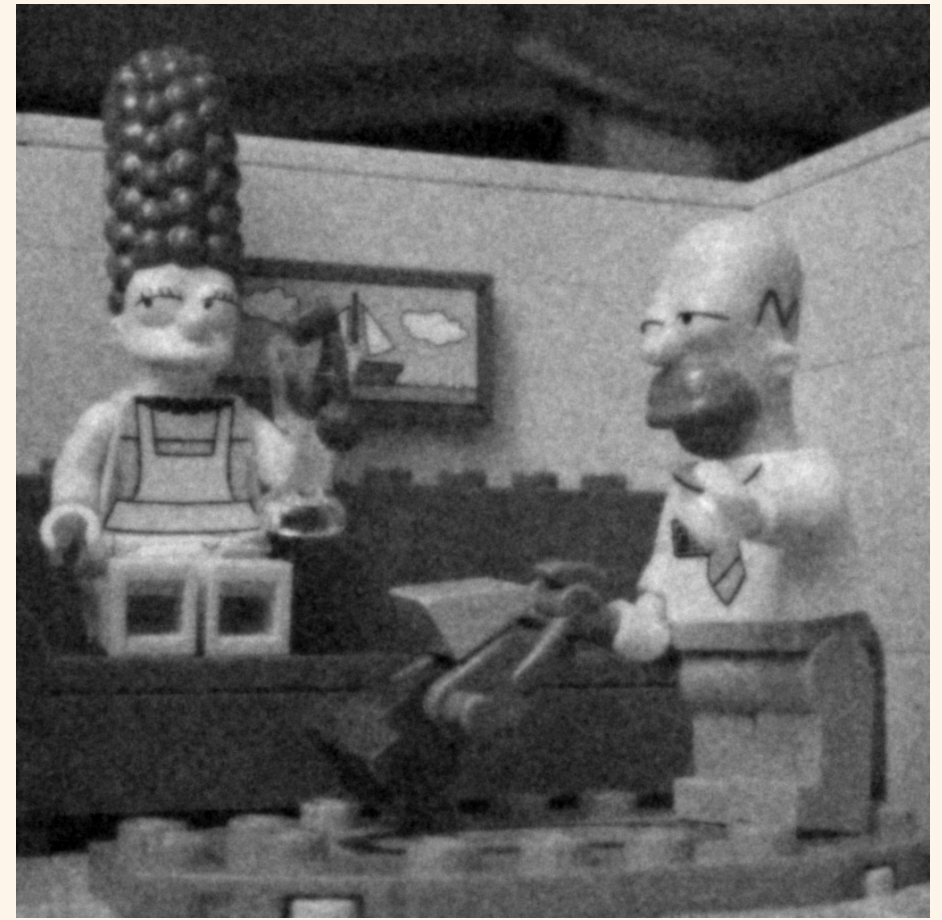
$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \lambda \sum_i \|\nabla x_i\|_2^2}_{E(x)}$$

convex, differentiable, gradient descent with

$$\frac{\partial E}{\partial x} = A^*(Ax - y) - 2\lambda \Delta x$$



Ex: $A = Id$
 $N \sim \mathcal{N}(0, \sigma^2)$



Total variation

TV-L2 [Rudin-Osher-Fatemi 1992 for $A = \text{Id}$]

$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2}_{E(x)} + \lambda \text{TV}(x), \quad \text{with } \text{TV}(x) = \sum_i \|\nabla x_i\|$$

Total variation

TV-L2 [Rudin-Osher-Fatemi 1992 for $A = \text{Id}$]

$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2}_{E(x)} + \lambda \text{TV}(x), \text{ with } \text{TV}(x) = \sum_i \|\nabla x_i\|$$

convex, not differentiable (approx $\text{TV}(x)$ by $\sum \sqrt{\|\nabla x_i\|^2 + \epsilon^2}$)

$$\frac{\partial E}{\partial x} = A^*(Ax - y) - \lambda \text{div} \frac{\nabla x}{\|\nabla x\|}.$$

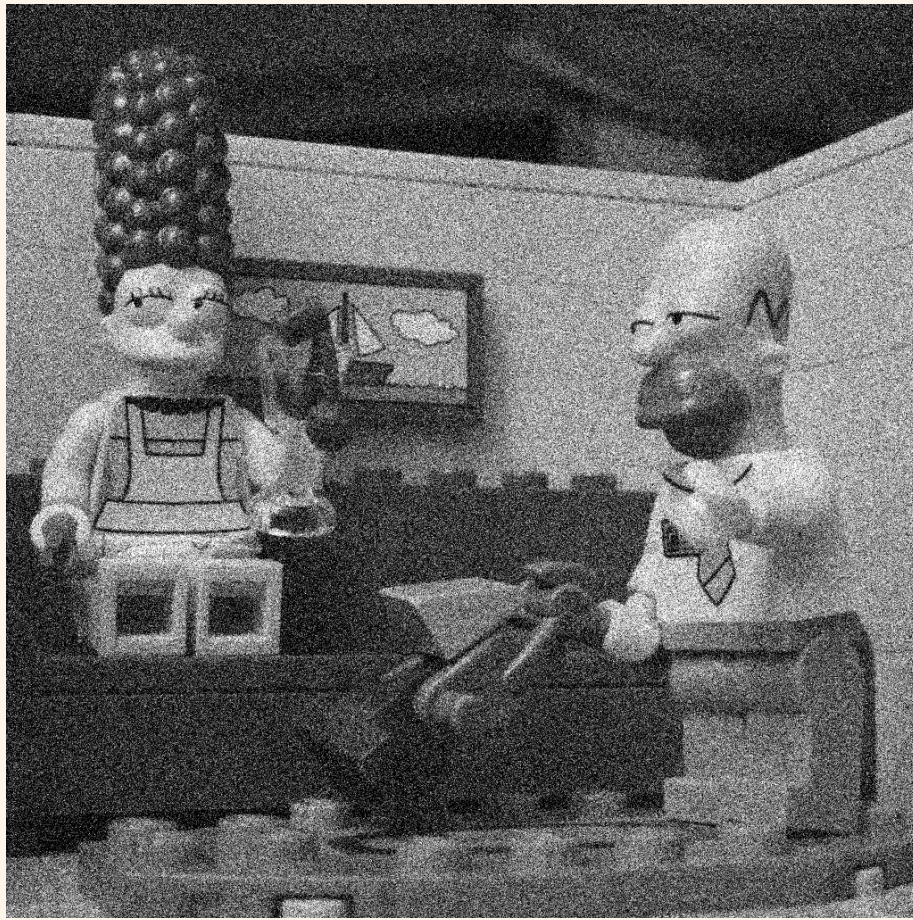
Total variation

TV-L2 [Rudin-Osher-Fatemi 1992 for $A = Id$]

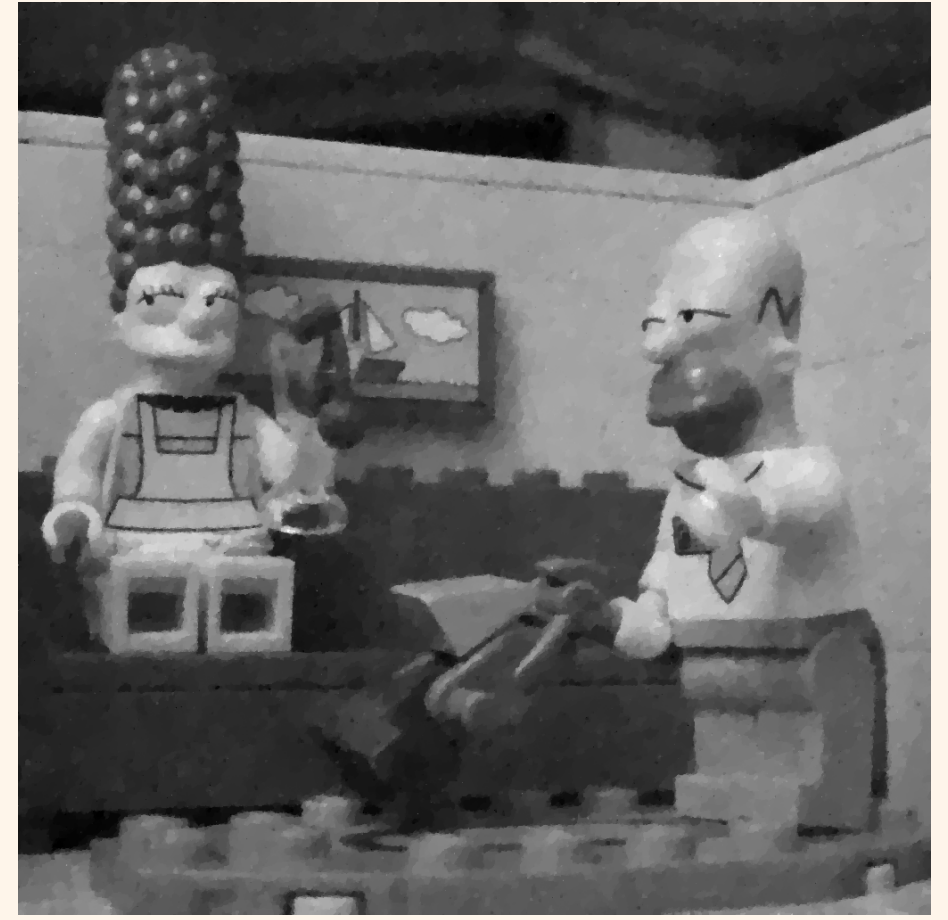
$$\arg \min_x \underbrace{\frac{1}{2} \|Ax - y\|_2^2}_{E(x)} + \lambda \text{TV}(x), \text{ with } \text{TV}(x) = \sum_i \|\nabla x_i\|$$

convex, not differentiable (approx $\text{TV}(x)$ by $\sum \sqrt{\|\nabla x_i\|^2 + \epsilon^2}$)

$$\frac{\partial E}{\partial x} = A^*(Ax - y) - \lambda \text{div} \frac{\nabla x}{\|\nabla x\|}.$$



Ex: $A = Id$
 $N \sim \mathcal{N}(0, \sigma^2)$



$$\operatorname{argmin}_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \operatorname{TV}(x) \quad (\mathcal{P})$$

TV-L2

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \operatorname{TV}(x) \quad (\mathcal{P})$$

Not differentiable

TV-L2

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \operatorname{TV}(x) \quad (\mathcal{P})$$

Let $X = \mathbb{R}^{n \times m}$, it happens that

Not differentiable

$$\operatorname{TV}(x) = \sum_k \|\nabla x(k)\|_2 = \max_{p \in X \times X} \langle \nabla x, p \rangle_X - \iota_{\|\cdot\|_{\infty,2}}(p)$$

$$\text{with } \iota_{\|\cdot\|_{\infty,i}}(p) = \begin{cases} 0 & \text{if } \|p\|_{\infty,2} := \sup_k \|p(k)\|_2 \leq 1 \\ +\infty & \text{otherwise.} \end{cases}$$

TV-L2

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \operatorname{TV}(x) \quad (\mathcal{P})$$

Let $X = \mathbb{R}^{n \times m}$, it happens that

Not differentiable

$$\operatorname{TV}(x) = \sum_k \|\nabla x(k)\|_2 = \max_{p \in X \times X} \langle \nabla x, p \rangle_X - \iota_{\|\cdot\|_{\infty,2}}(p)$$

$$\text{with } \iota_{\|\cdot\|_{\infty,i}}(p) = \begin{cases} 0 & \text{if } \|p\|_{\infty,2} := \sup_k \|p(k)\|_2 \leq 1 \\ +\infty & \text{otherwise.} \end{cases}$$

Thus, $(\mathcal{P})$ can be rewritten as a saddle point problem

$$\operatorname{argmin}_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,i}}(p)}_{H(x,p)}$$

TV-L2

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \operatorname{TV}(x) \quad (\mathcal{P})$$

Let $X = \mathbb{R}^{n \times m}$, it happens that

Not differentiable

$$\operatorname{TV}(x) = \sum_k \|\nabla x(k)\|_2 = \max_{p \in X \times X} \langle \nabla x, p \rangle_X - \iota_{\|\cdot\|_{\infty,2}}(p)$$

$$\text{with } \iota_{\|\cdot\|_{\infty,i}}(p) = \begin{cases} 0 & \text{if } \|p\|_{\infty,2} := \sup_k \|p(k)\|_2 \leq 1 \\ +\infty & \text{otherwise.} \end{cases}$$

Thus, $(\mathcal{P})$ can be rewritten as a saddle point problem

$$\operatorname{arg min}_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,i}}(p)}_{H(x,p)}$$

Not differentiable

Back to TV-L2

$$\arg \min_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,2}}(p)}_{H(x,p)}$$

Back to TV-L2

$$\arg \min_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,2}}(p)}_{H(x,p)}$$

Chambolle-Pock (2010)

$H(x,p)$

x_0, p_0 initialization

$$p_{k+1} = \text{prox}_{-\tau H_{x_k}}(p_k) = \pi_{\|\cdot\|_{\infty,2}}(p_k + \tau \lambda \nabla \bar{x}_k)$$

$$\begin{aligned} x_{k+1} &= \text{prox}_{\sigma H_{p_{k+1}}}(x_k) \\ &= (Id + \sigma A^* A)^{-1} (x_k + \sigma A^* y + \lambda \sigma \text{div}(p_k)) \end{aligned}$$

$$\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k)$$

Back to TV-L2

$$\arg \min_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,2}}(p)}_{H(x,p)}$$

Chambolle-Pock (2010)

$H(x,p)$

x_0, p_0 initialization

$$p_{k+1} = \text{prox}_{-\tau H_{x_k}}(p_k) = \pi_{\|\cdot\|_{\infty,2}}(p_k + \tau \lambda \nabla \bar{x}_k)$$

$$\begin{aligned} x_{k+1} &= \text{prox}_{\sigma H_{p_{k+1}}}(x_k) \\ &= (Id + \sigma A^* A)^{-1} (x_k + \sigma A^* y + \lambda \sigma \text{div}(p_k)) \end{aligned}$$

$$\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k)$$

Makes sense because prox_{H_p} and prox_{H_u} are easy to compute !!!

convergence if $8\sigma\tau\lambda^2 < 1$

Back to TV-L2

$$\arg \min_x \max_p \underbrace{\frac{1}{2} \|Ax - y\|_2^2 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,2}}(p)}_{H(x,p)}$$

Chambolle-Pock (2010)

$H(x,p)$

x_0, p_0 initialization

$$p_{k+1} = \text{prox}_{-\tau H_{x_k}}(p_k) = \pi_{\|\cdot\|_{\infty,2}}(p_k + \tau \lambda \nabla \bar{x}_k)$$

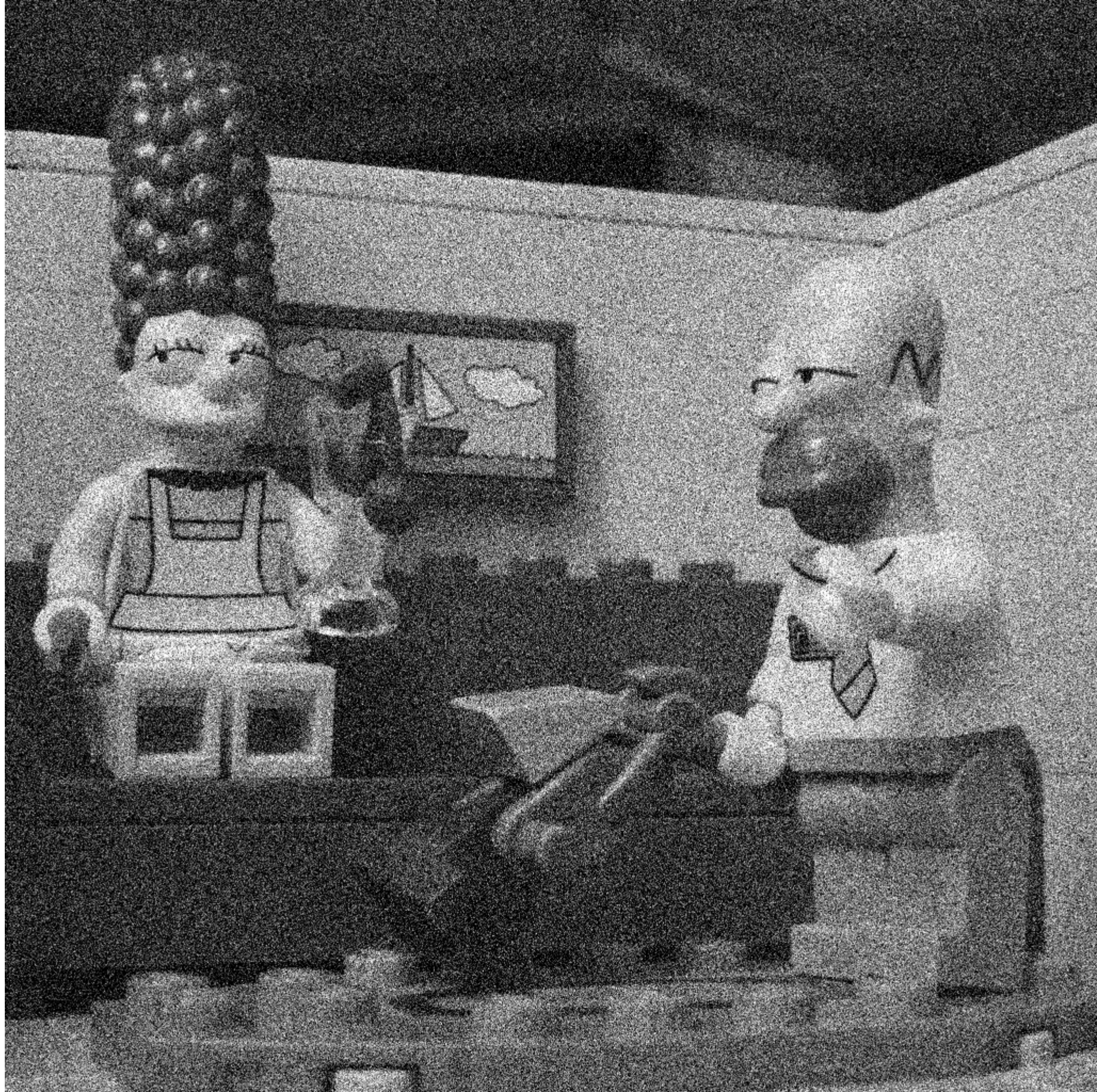
$$\begin{aligned} x_{k+1} &= \text{prox}_{\sigma H_{p_{k+1}}}(x_k) \\ &= (Id + \sigma A^* A)^{-1} (x_k + \sigma A^* y + \lambda \sigma \text{div}(p_k)) \end{aligned}$$

$$\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k)$$

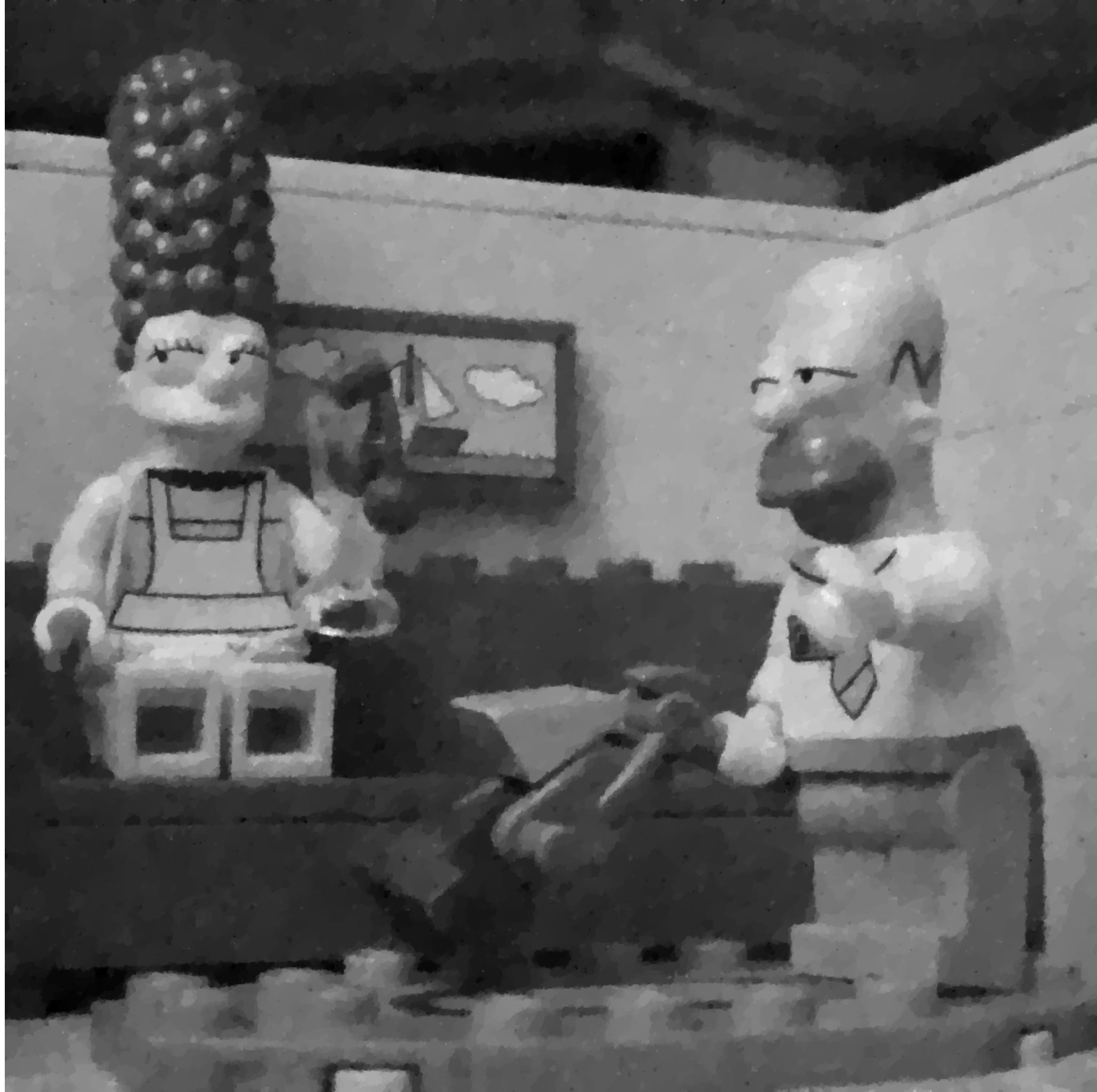
Makes sense because prox_{H_p} and prox_{H_u} are easy to compute !!!

See practical session on
Image restoration !

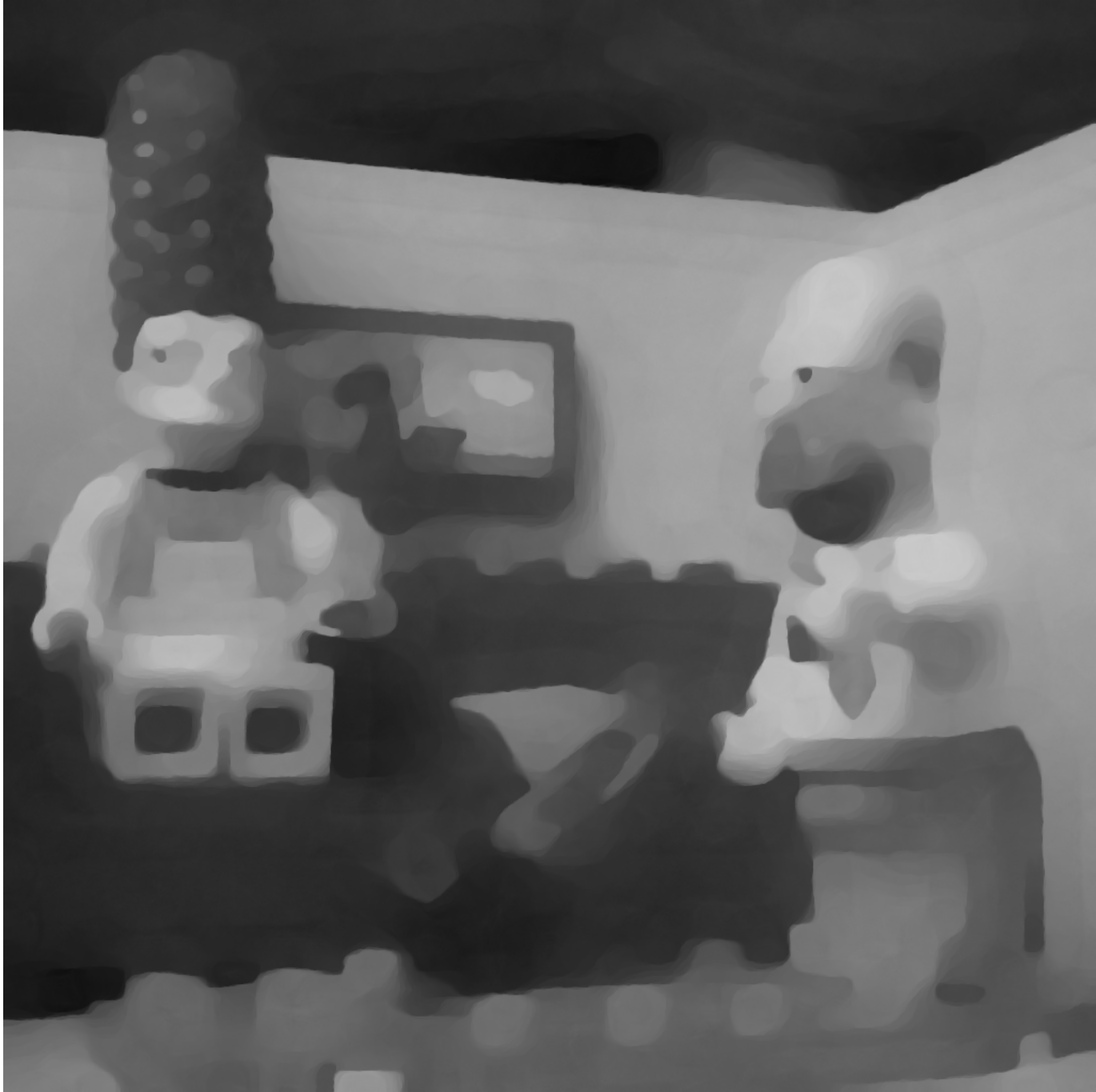
convergence if $8\sigma\tau\lambda^2 < 1$



Gaussian
noise
 $\sigma = 0.2$



TV-L2
 λ large



Wavelet denoising

$$\operatorname{argmin}_x \|x - y\|_2^2 + \lambda \|Wx\|_1 \quad (\mathcal{P}')$$

If W is an orthonormal basis, pb equivalent to

$$\operatorname{argmin}_x \|Wx - Wy\|_2^2 + \lambda \|Wx\|_1 \quad (\mathcal{P}')$$

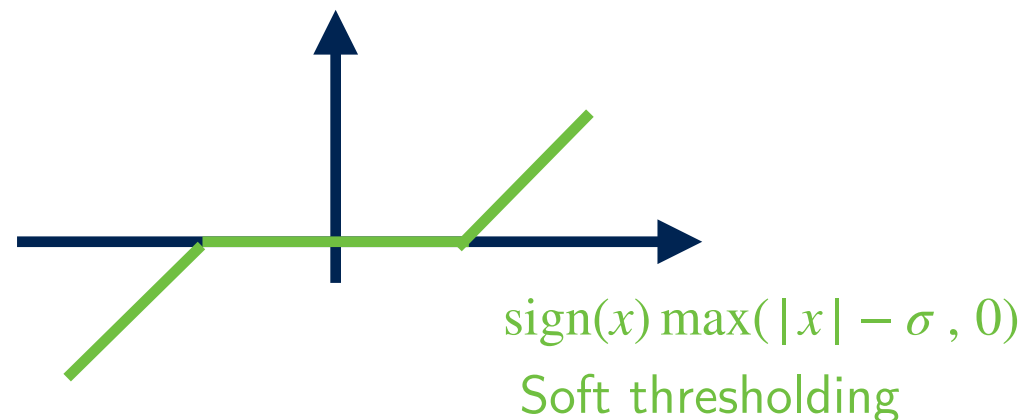
Wavelet denoising

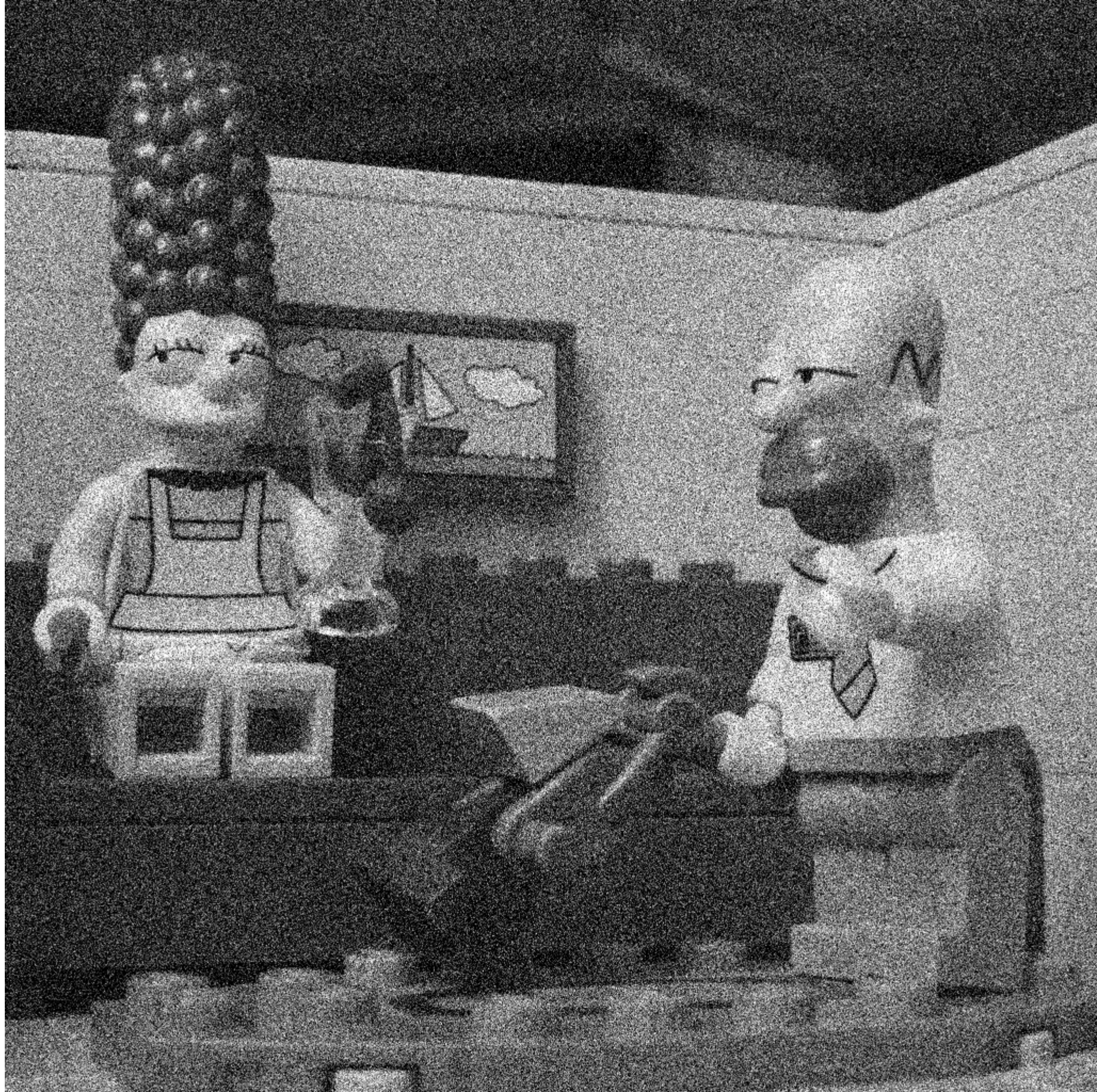
$$\operatorname{argmin}_x \|x - y\|_2^2 + \lambda \|Wx\|_1 \quad (\mathcal{P}')$$

If W is an orthonormal basis, pb equivalent to

$$\operatorname{argmin}_x \|Wx - Wy\|_2^2 + \lambda \|Wx\|_1 \quad (\mathcal{P}')$$

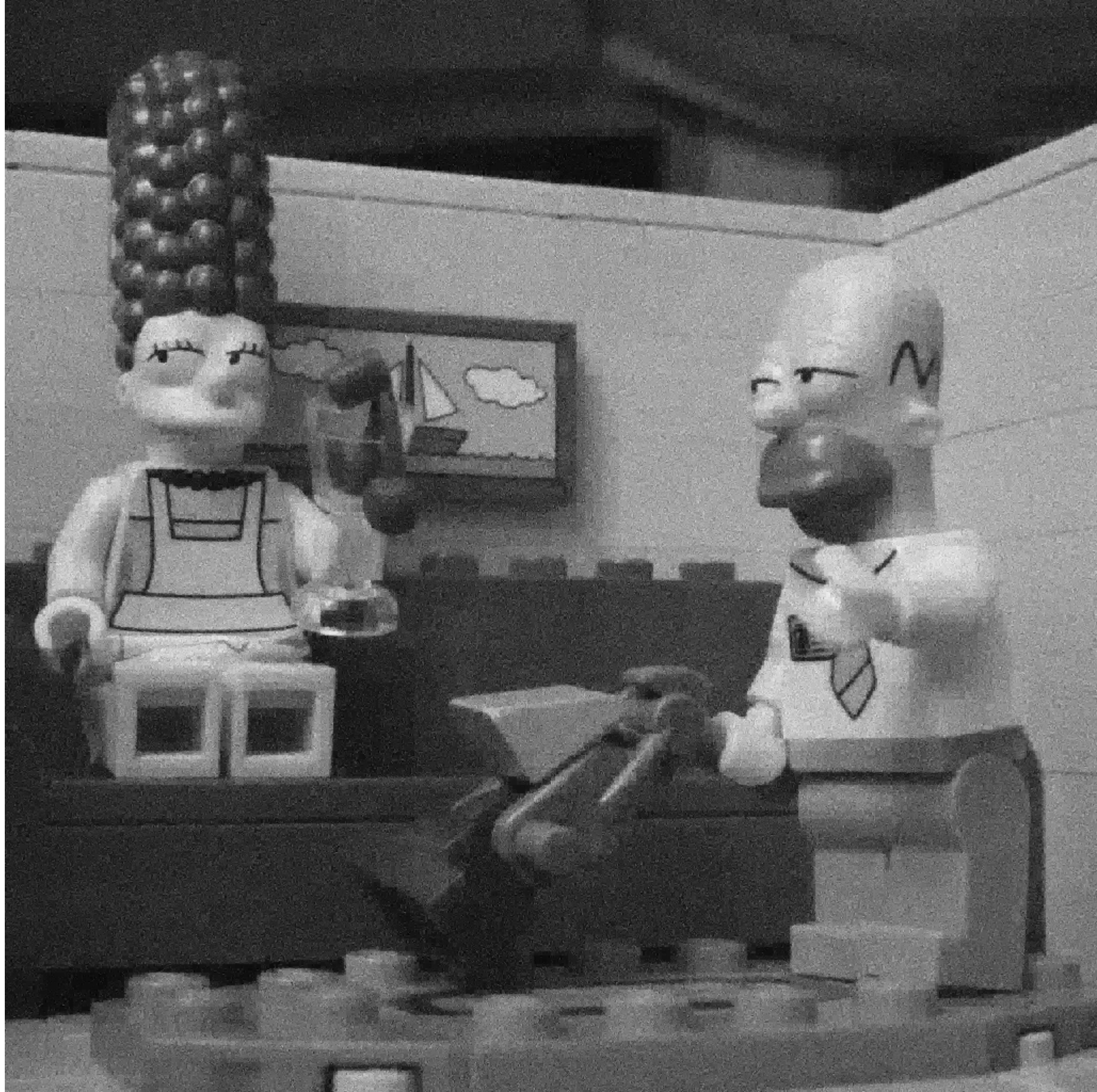
Soft thresholding in the Wavelet domain...





Gaussian
noise
 $\sigma = 0.2$

Wavelet
denoising



Progress in image denoising



Noisy image with $\sigma = 50$

Progress in image denoising



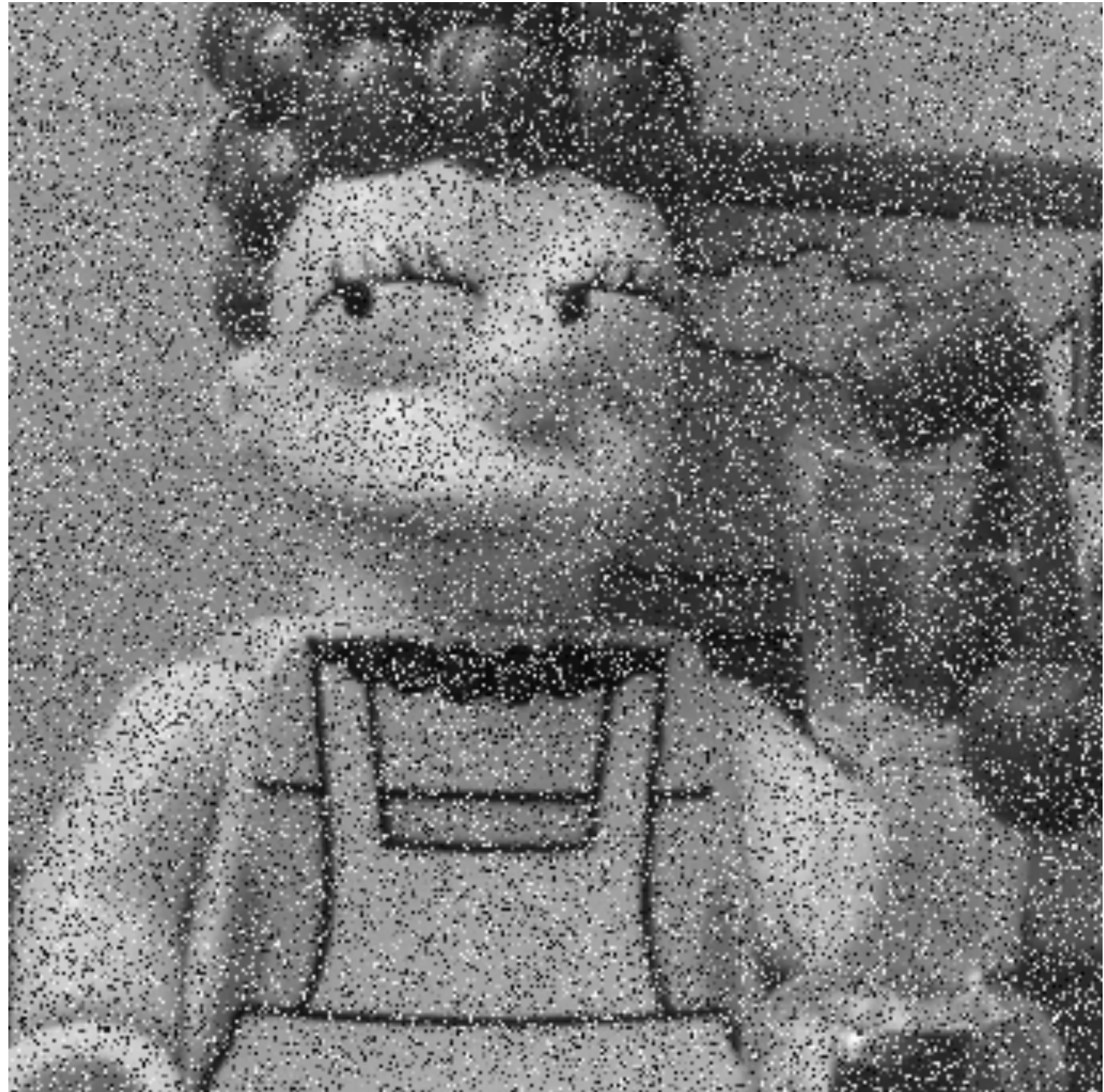
TV-L2 denoising with optimal λ

Progress in image denoising



Wavelet denoising

What about impulse noise ?



TV-L1 for impulse noise

$$\operatorname{argmin}_x \|x - y\|_1 + \lambda \operatorname{TV}(x) \quad (\mathcal{P}')$$

$(\mathcal{P}')$ can be rewritten as

$$\operatorname{arg min}_x \max_p \underbrace{\|x - y\|_1 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_\infty, i}(p)}_{H(x, p)}$$

TV-L1 for impulse noise

$$\operatorname{argmin}_x \|x - y\|_1 + \lambda \operatorname{TV}(x) \quad (\mathcal{P}')$$

$(\mathcal{P}')$ can be rewritten as

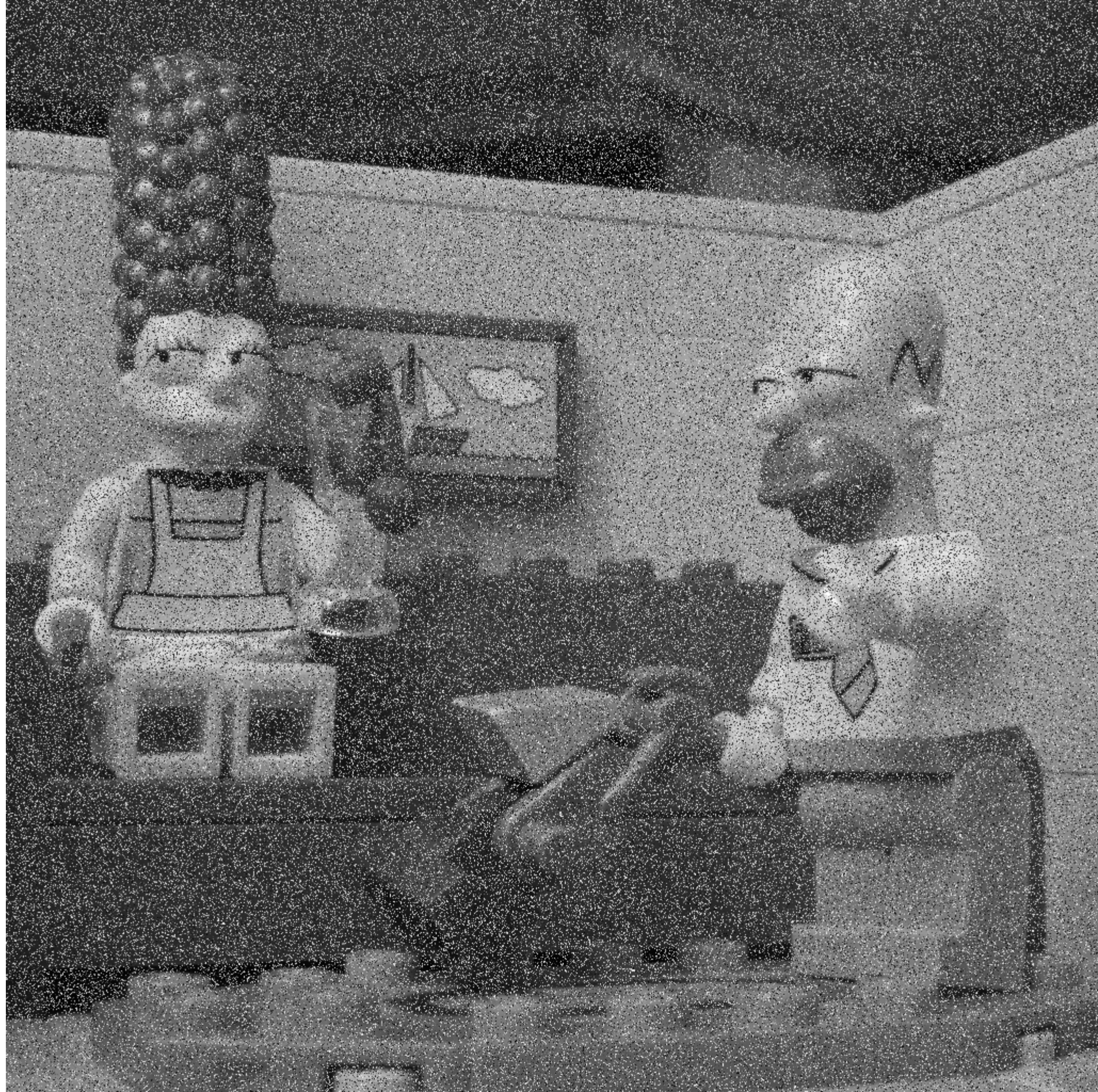
$$\operatorname{argmin}_x \max_p \underbrace{\|x - y\|_1 + \langle \lambda \nabla x, p \rangle - \iota_{\|\cdot\|_{\infty,i}}(p)}_{H(x,p)}$$

Algorithm

x_0, p_0 initialization

$$x_{k+1} = \operatorname{prox}_{-\sigma H_{p_k}}(x_k) = \pi_{\|\cdot\|_{\infty,2}}(p_k + \tau \lambda \nabla x_k)$$

$$p_{k+1} = \operatorname{prox}_{\tau H_{x_{k+1}}}(p_k) = \text{soft thresh}_{\sigma} (x_k + \lambda \sigma \operatorname{div}(p_k) - y) + y$$



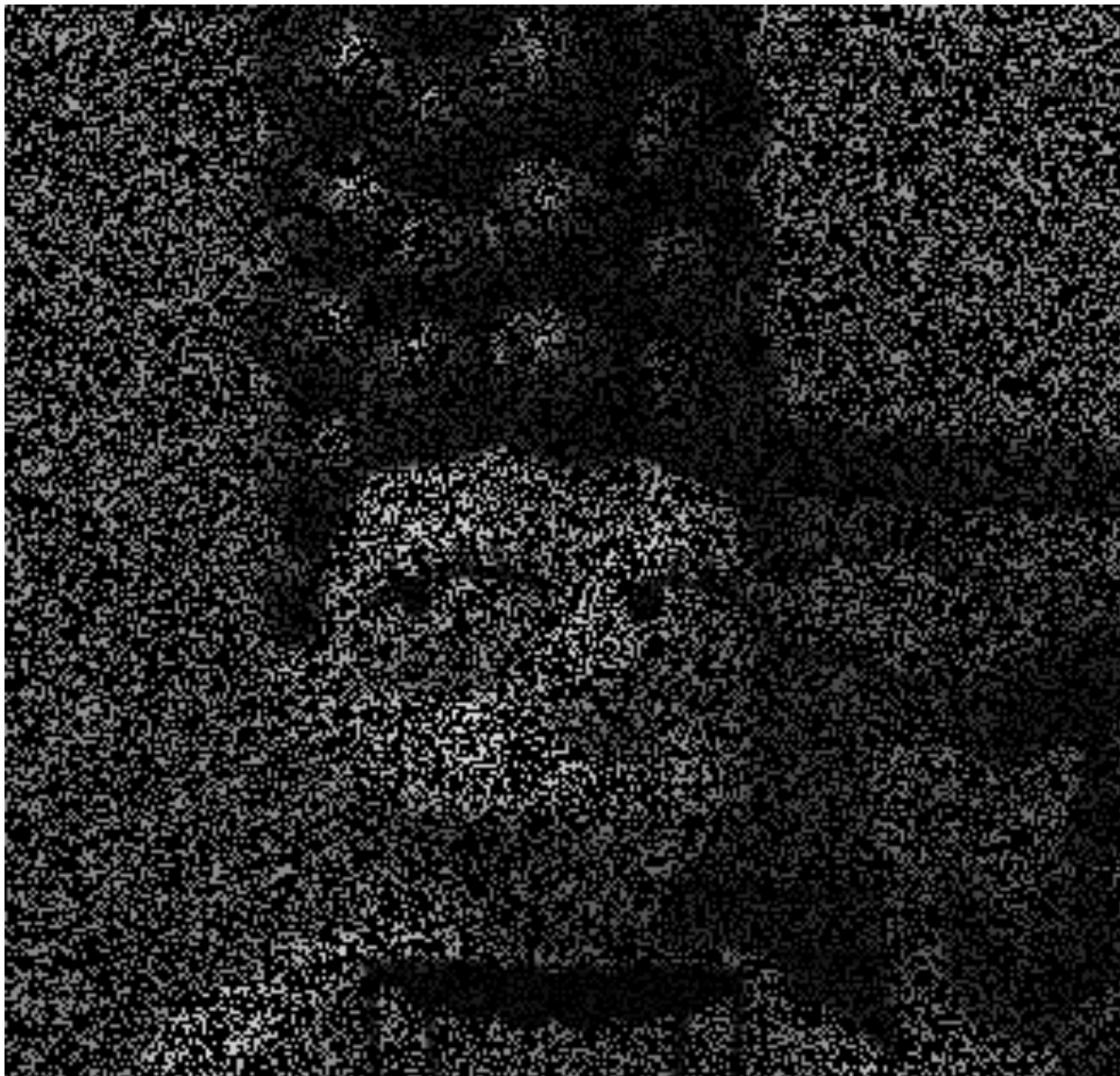
Impulse
noise
 $p = 0.3$



TV interpolation

$$\min_{x|_{mask}=y|_{mask}} \text{TV}(x) \quad (\mathcal{P}')$$

$$= \min_x \text{TV}(x) + \iota_C(x) \text{ with } C = \{x, x|_{mask} = y|_{mask}\}$$



Patch-based models

Multi-image denoising: centered noise

Input: burst of registered images $X_1, \dots, X_n$

s.t. $X_i = X + B_i$ with B_i i.i.d. centered

Output: average $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{n \rightarrow +\infty} X$



+



+...+

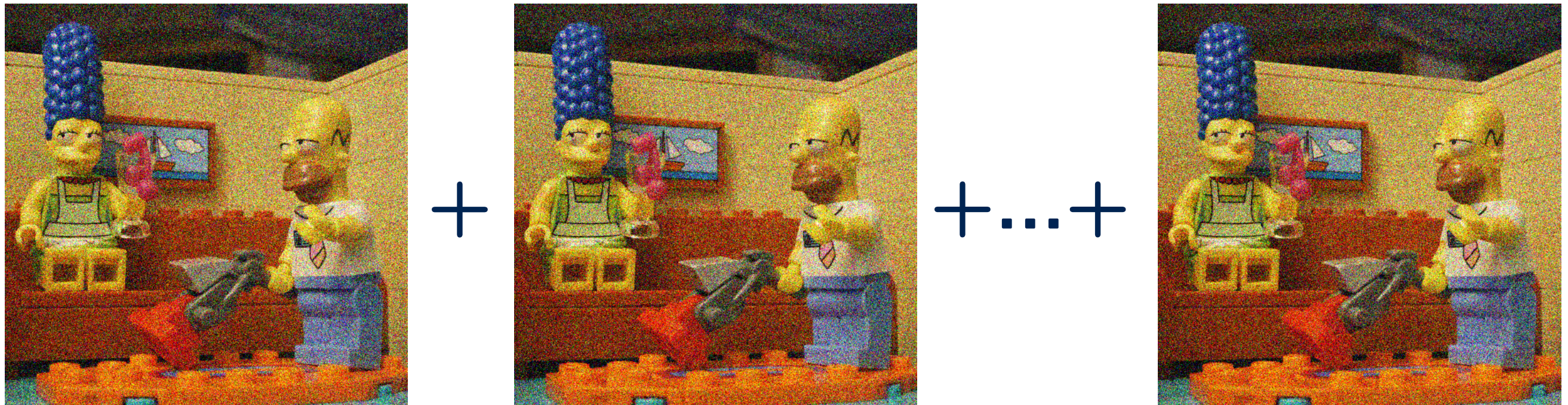


Multi-image denoising: centered noise

Input: burst of registered images $X_1, \dots, X_n$

s.t. $X_i = X + B_i$ with B_i i.i.d. centered

Output: average $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{n \rightarrow +\infty} X$



Noise reduction by a factor $\frac{1}{\sqrt{n}}$

Simpsons, $\sigma = 0.1$



Simpsons, $\sigma = 0.1$, mean of 5 images



Simpsons, $\sigma = 0.1$, mean of 10 images



Simpsons, $\sigma = 0.1$, mean of 40 images



Simpsons, original image



Non local Means (Buadès, Coll, Morel 2005)

image x



pixel i

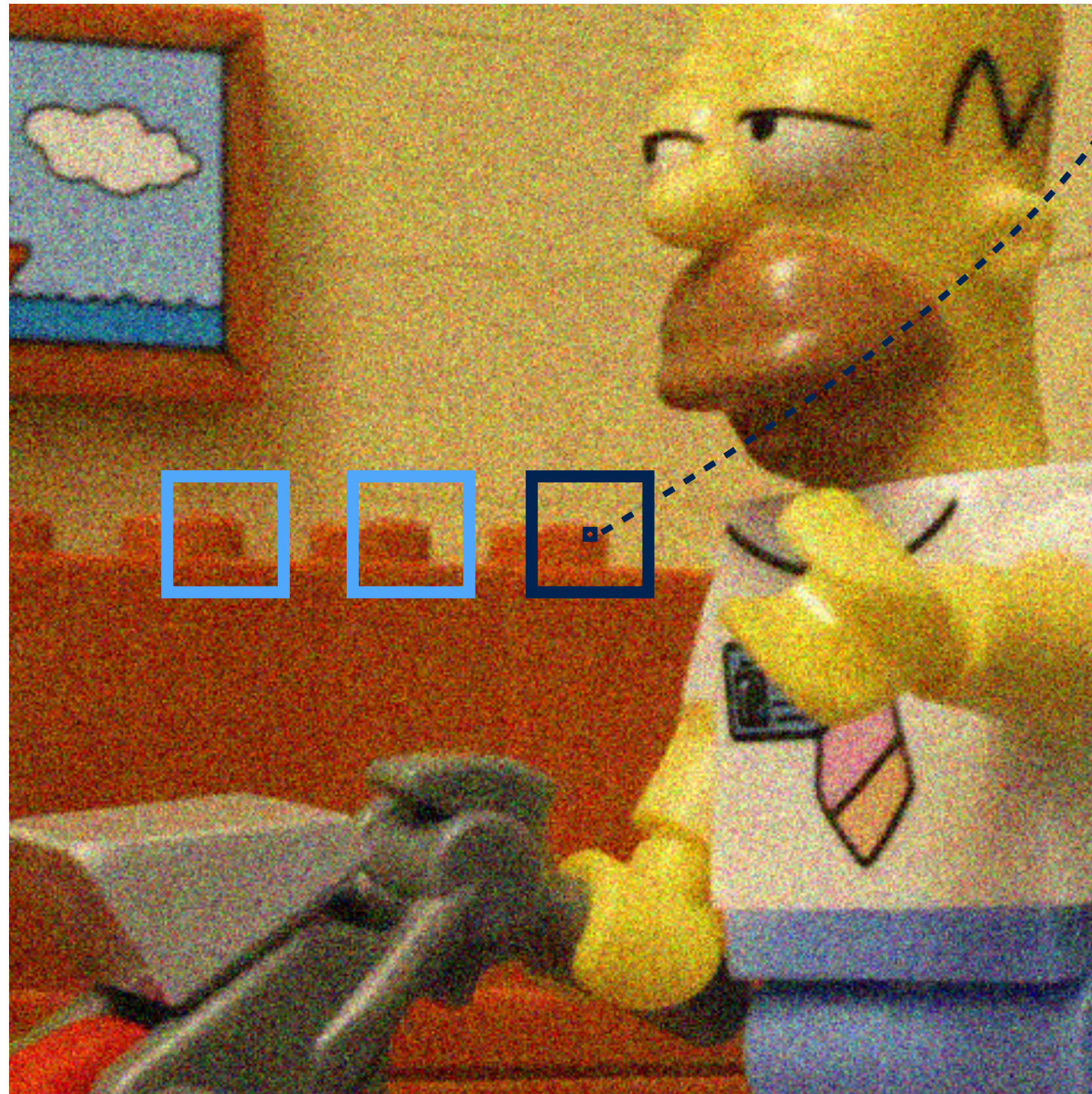
Non local Means (Buadès, Coll, Morel 2005)

image x



Non local Means (Buadès, Coll, Morel 2005)

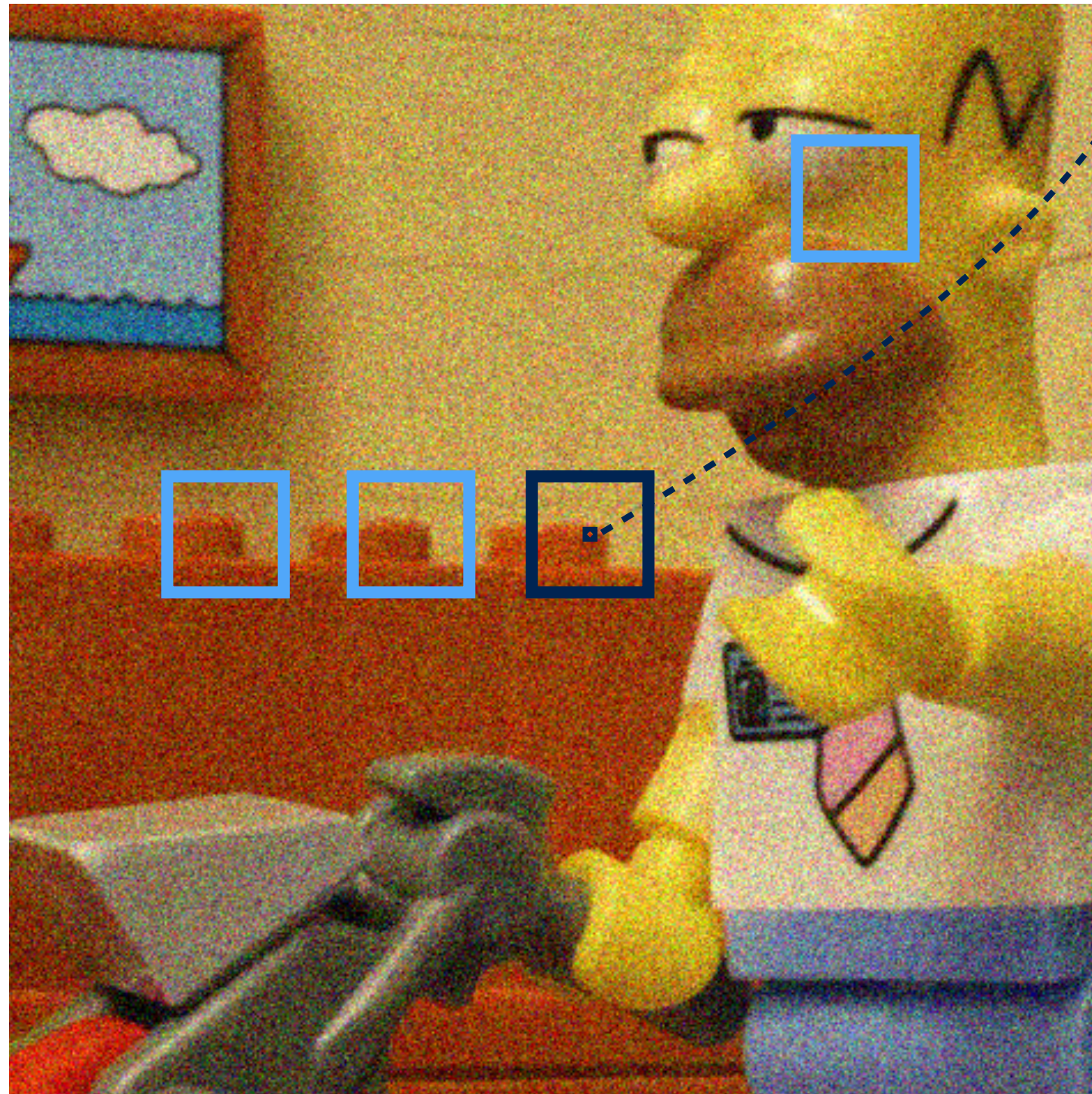
image x



pixel i

Non local Means (Buadès, Coll, Morel 2005)

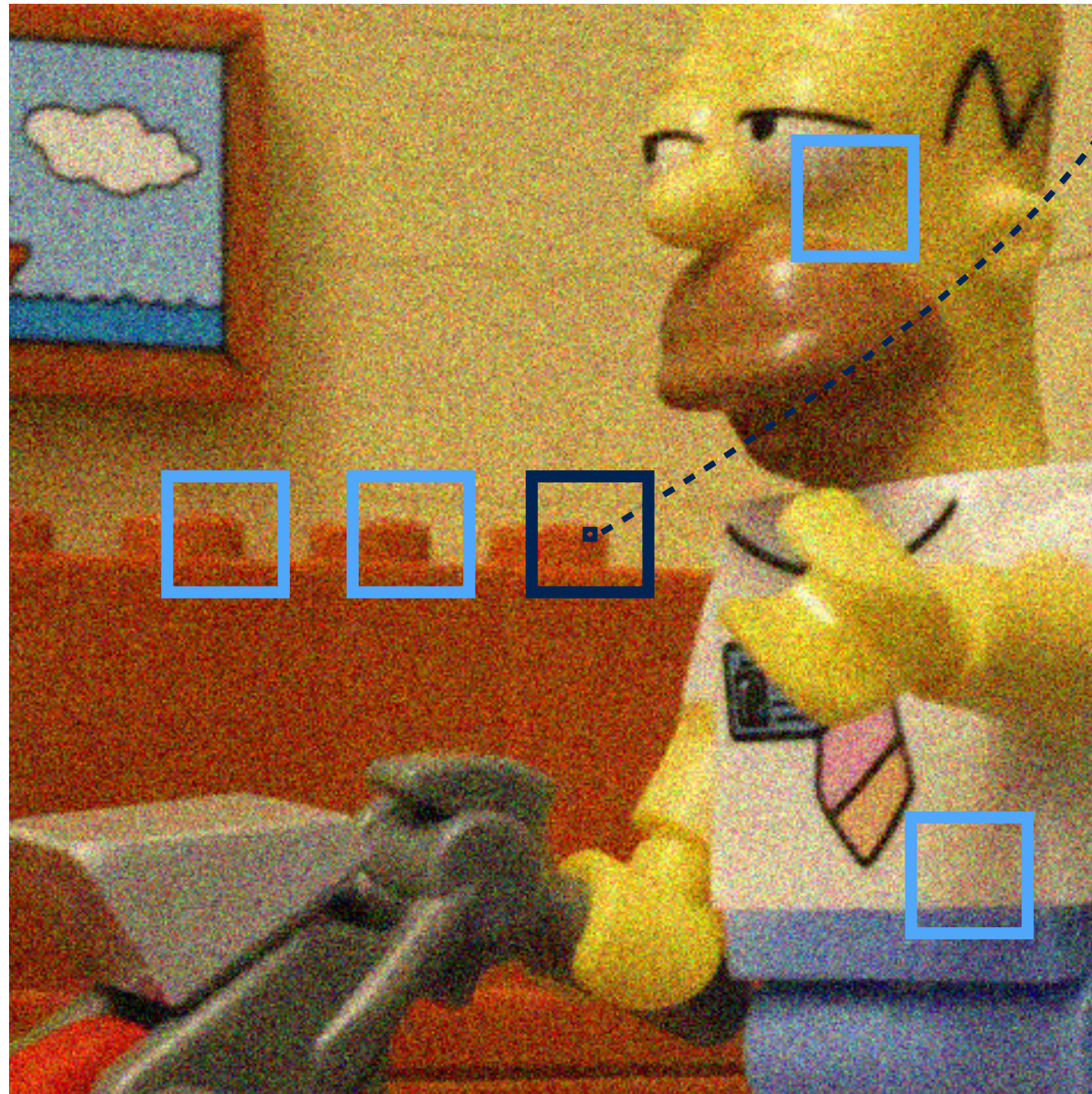
image x



pixel i

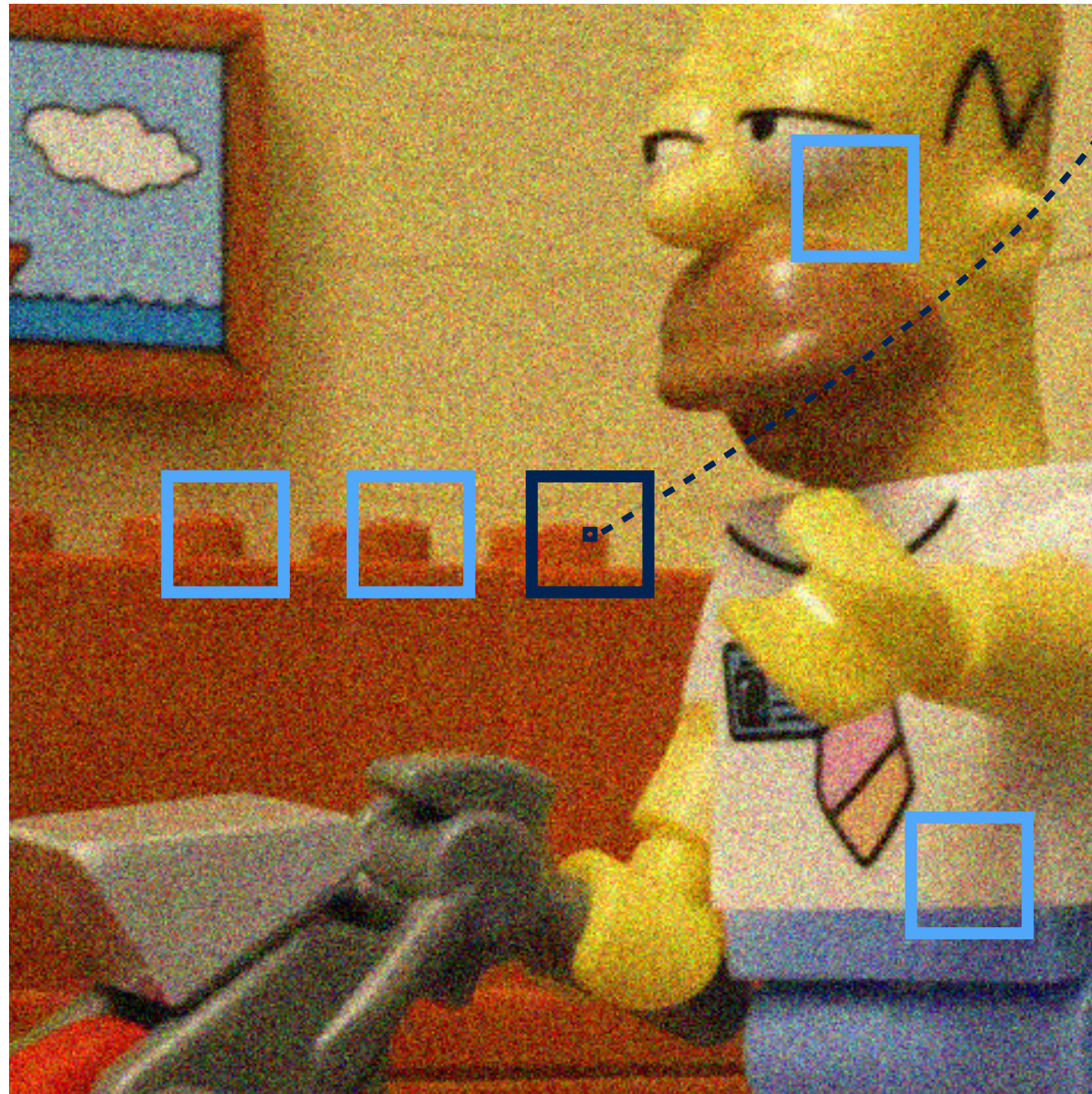
Non local Means (Buadès, Coll, Morel 2005)

image x



Non local Means (Buadès, Coll, Morel 2005)

image x



pixel i

$x(i)$ denoised = $\frac{1}{\sum_l w_{i,l}} \sum_k w_{i,k} x(k)$ with $w_{i,k}$ positive weights,
functions of the similarity between patches centered at i and k .

Non local Means (Buadès, Coll, Morel 2005)



Non local Means (Buadès, Coll, Morel 2005)

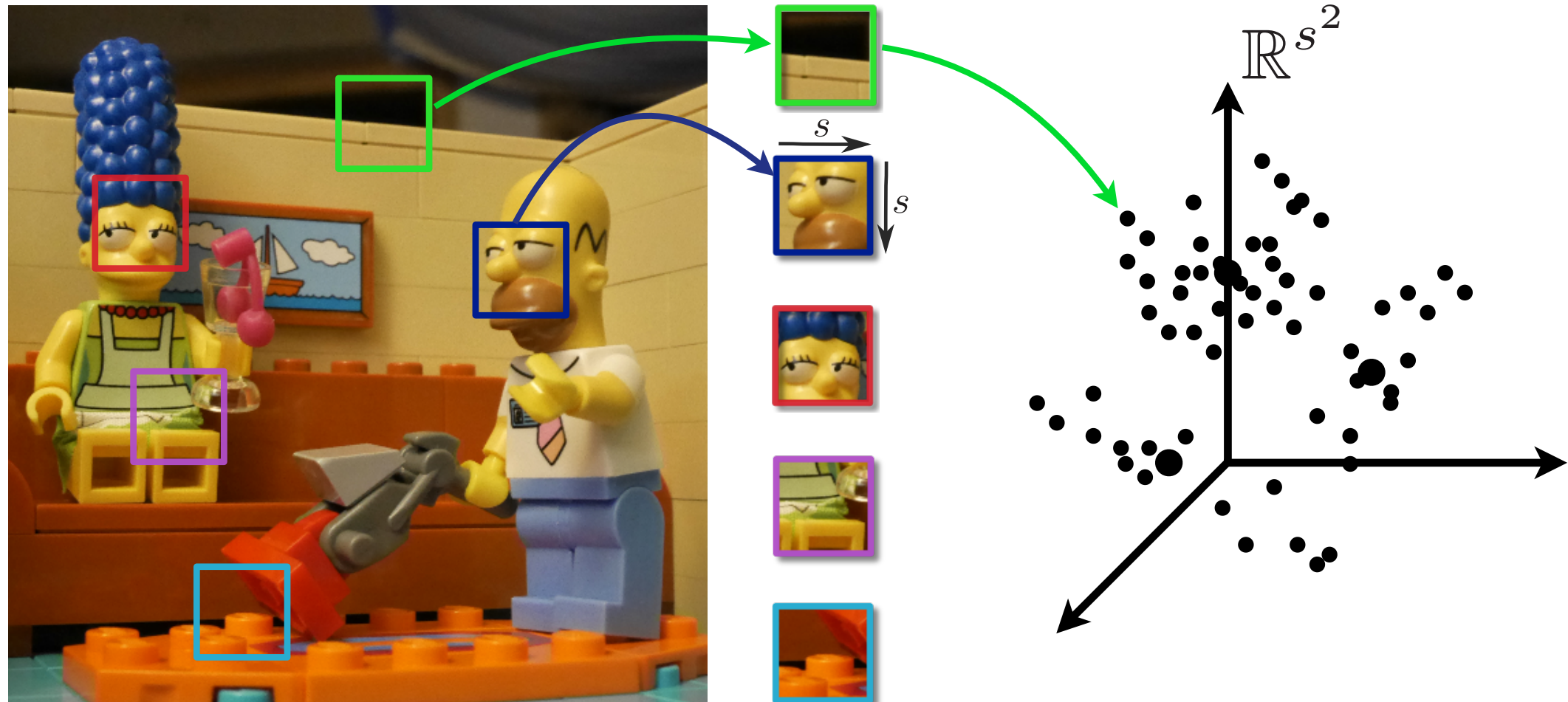


Original image

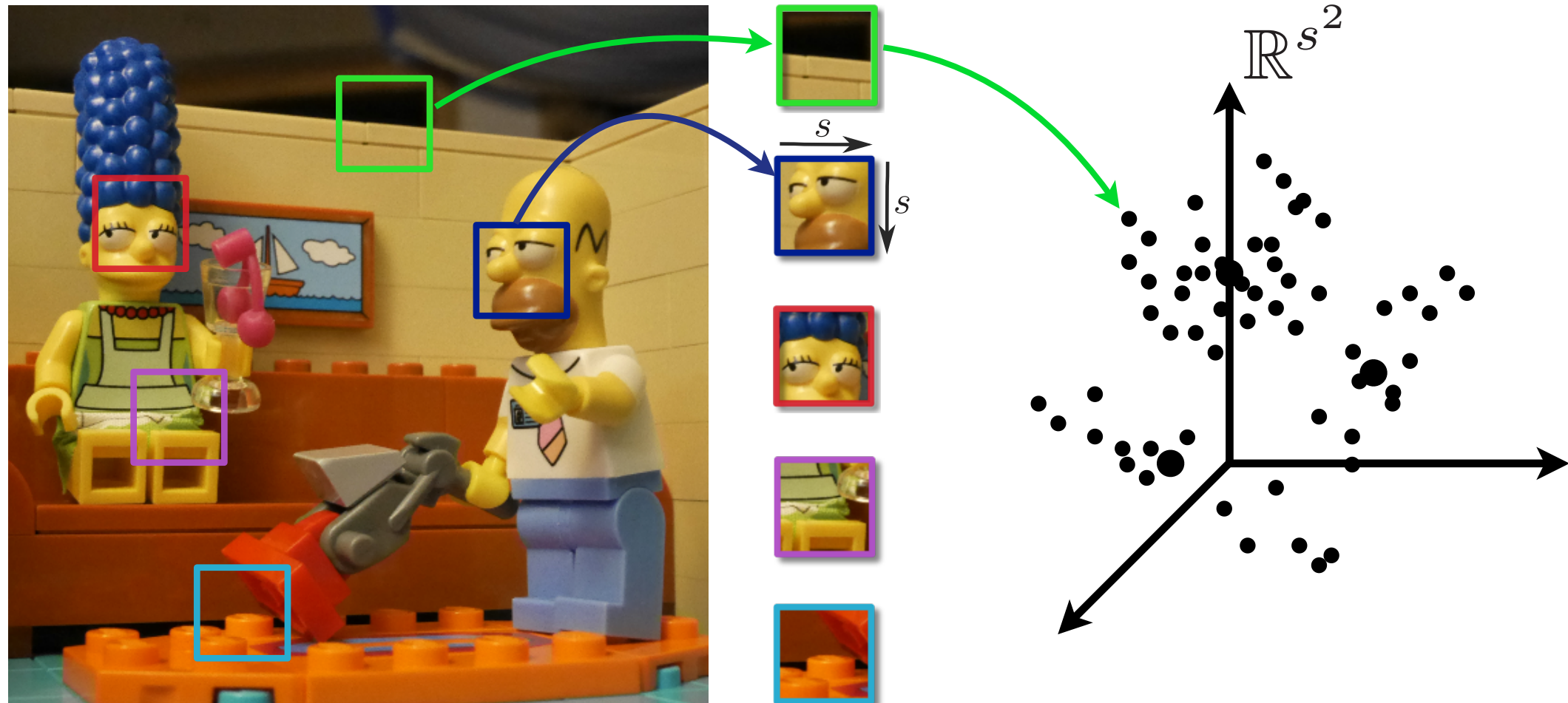


http://www.ipol.im/pub/art/2011/bcm_nlm/

Patch-based models and bayesian framework

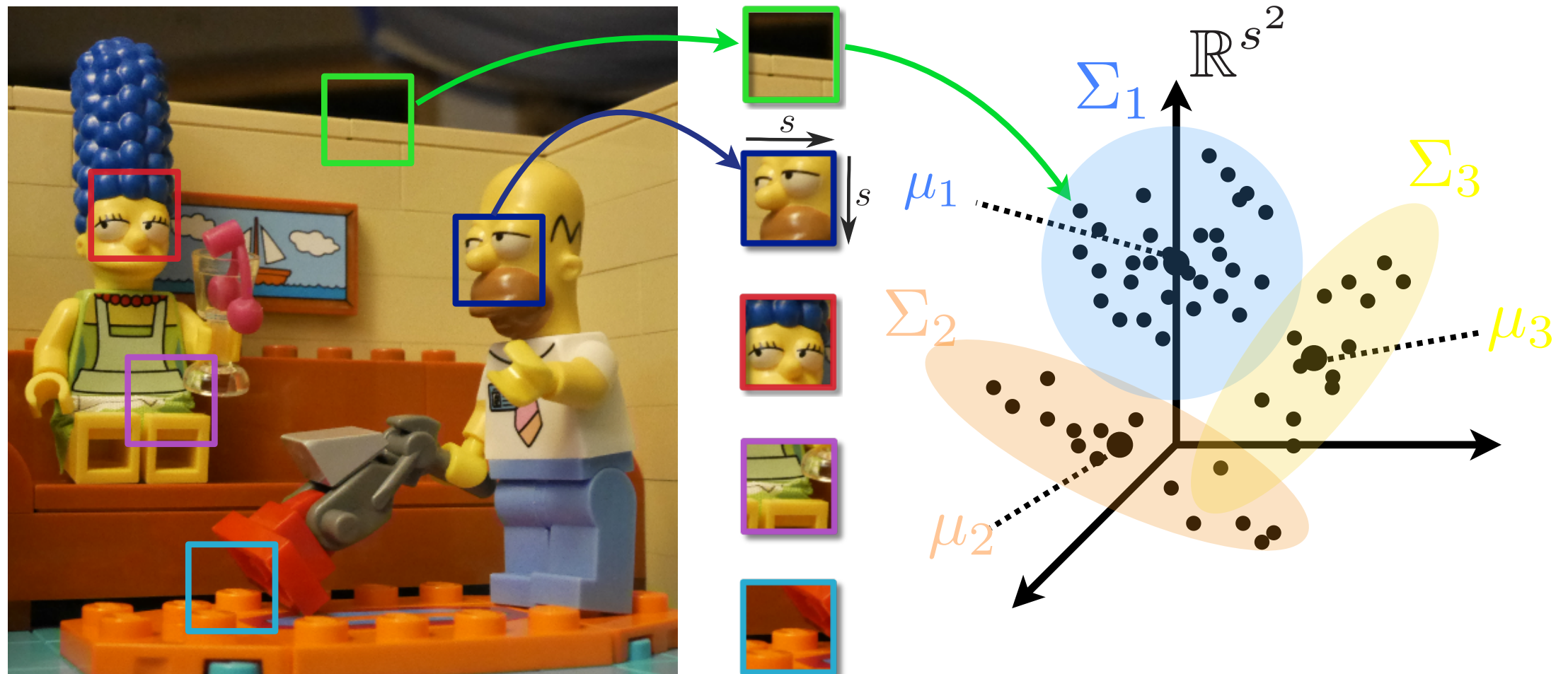


Patch-based models and bayesian framework



- Patch transform (ACP, DCT, dictionnaires) and adaptive thresholding

Patch-based models and bayesian framework



- Patch transform (ACP, DCT, dictionnaires) and adaptive thresholding
- Explicit probability models on patches: $\mathcal{N}(\mu, \Sigma)$ or $\sum_{k=1}^K \pi_k \mathcal{N}(\mu_k, \Sigma_k)$

Gaussian models and GMM on patches

MAP = MMSE for Gaussian measures

Proposition: Assume $X \sim \mathcal{N}(\mu, \Sigma)$ and $Y|X = x \sim \mathcal{N}(Ax, \sigma^2 I)$, then

$$\mathbb{E}[X|Y = y] = \arg \max_x \log p[X = x|Y = y] = \psi_k(y)$$

$$\text{with } \psi_k(y) := \mu + \Sigma A^t (A \Sigma A^t + \sigma^2 I)^{-1} (y - A \mu).$$

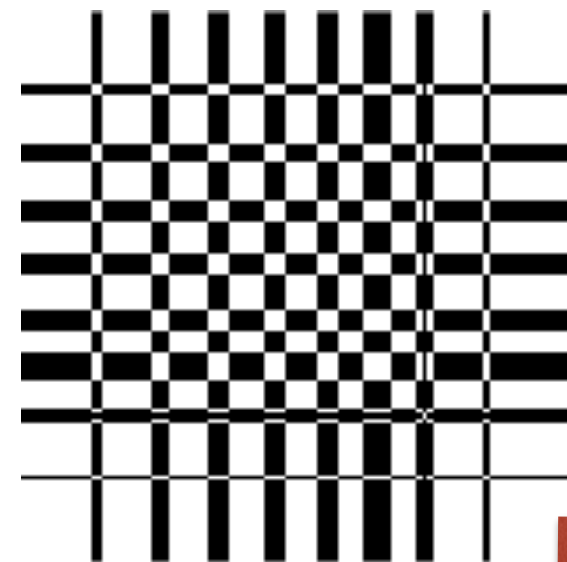
Proposition: Assume $X \sim \sum_{k=1}^K \pi_k \mathcal{N}(\mu_k; \Sigma_k)$ and $Y|X = x \sim \mathcal{N}(Ax, \sigma^2 I)$, then

$$\mathbb{E}[X|Y = y] = \sum_{k=1}^K \pi_k \psi_k(y).$$

Gaussian priors on patches



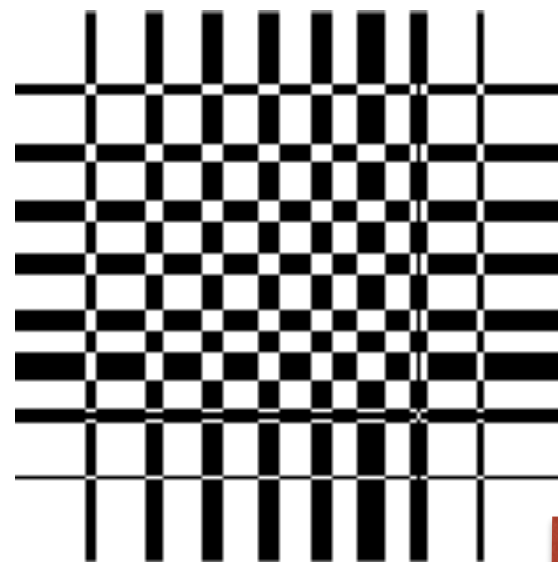
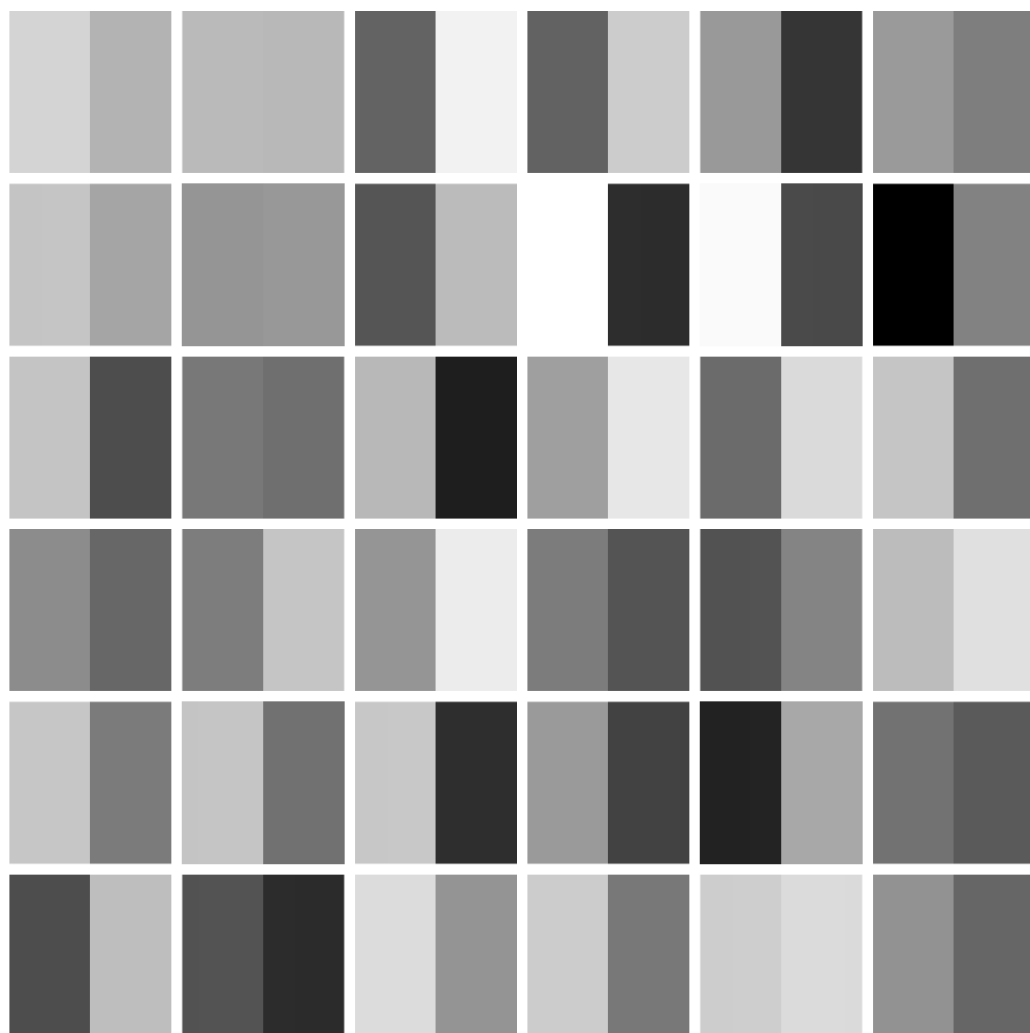
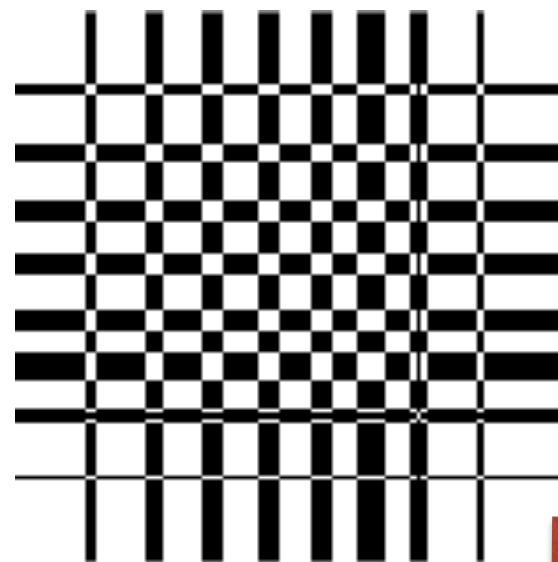
covariance matrices



Gaussian priors on patches



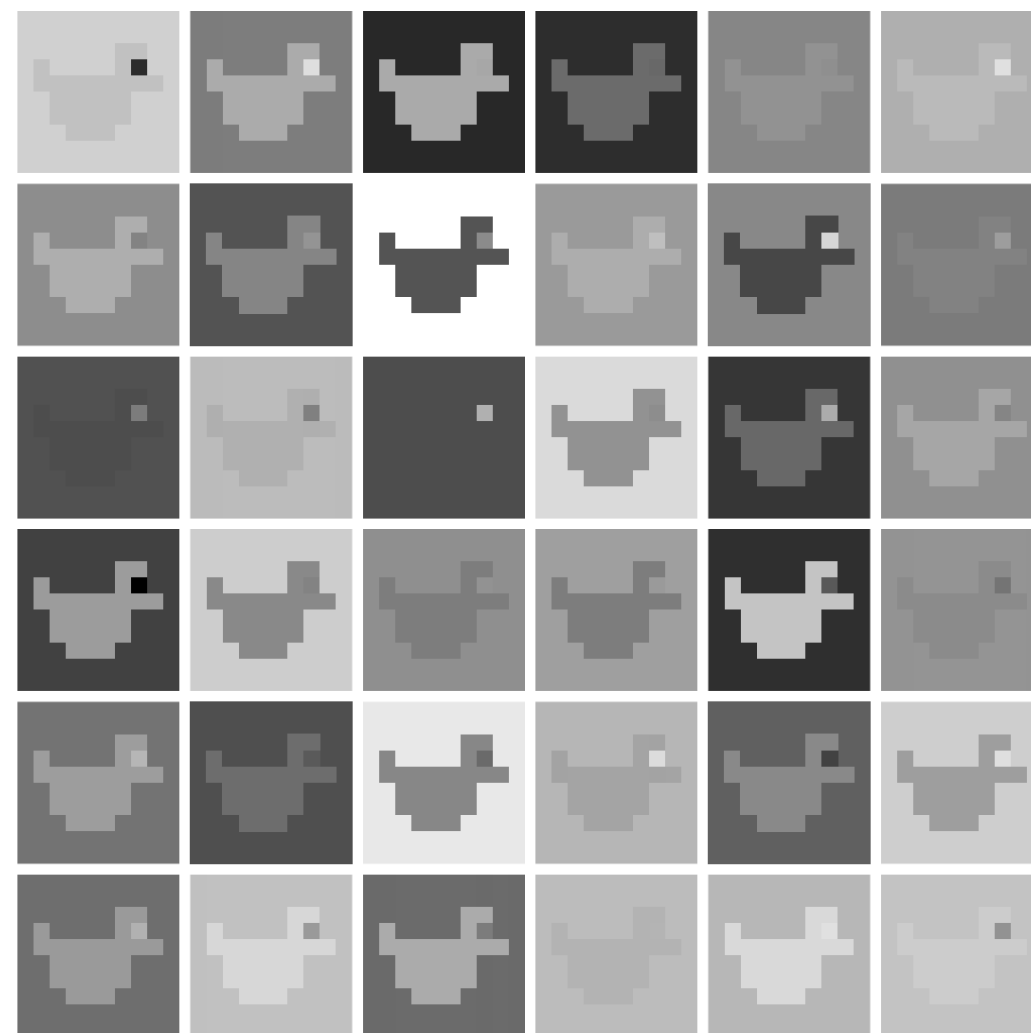
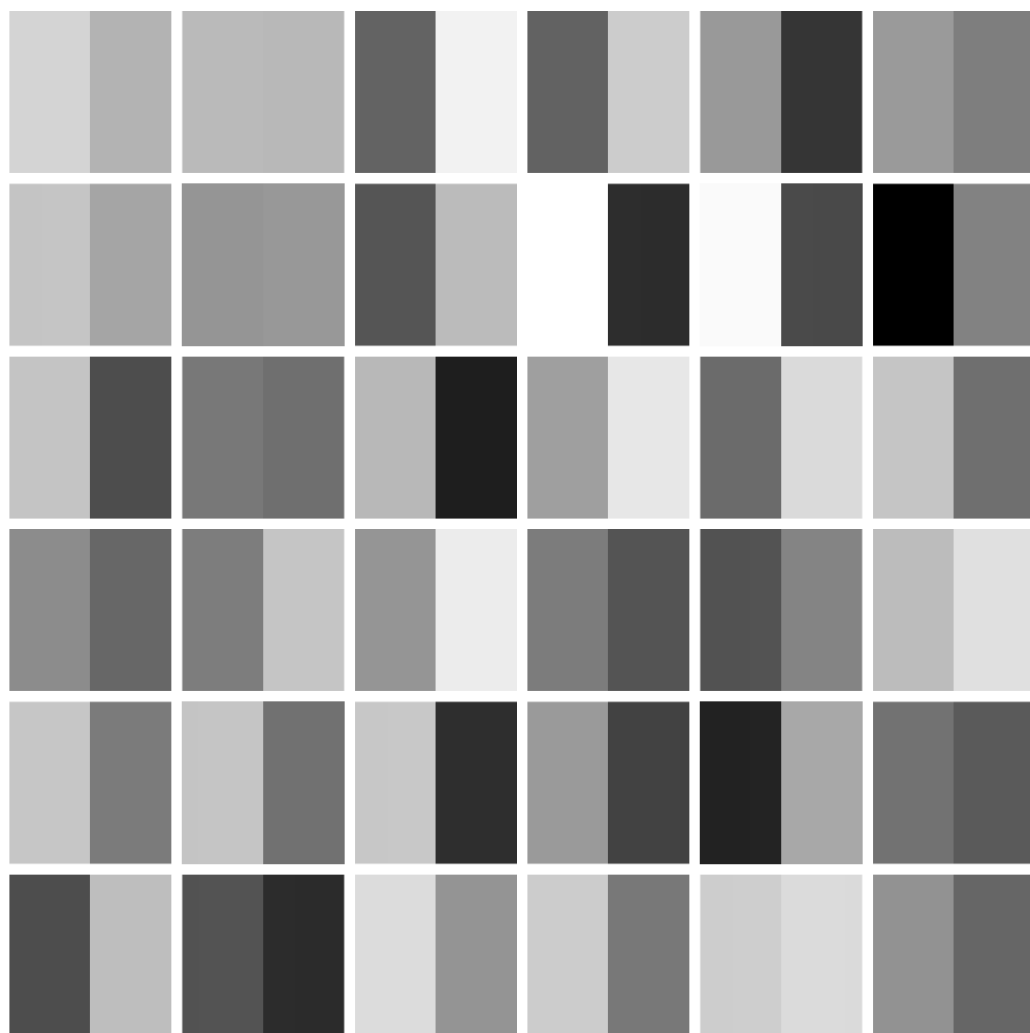
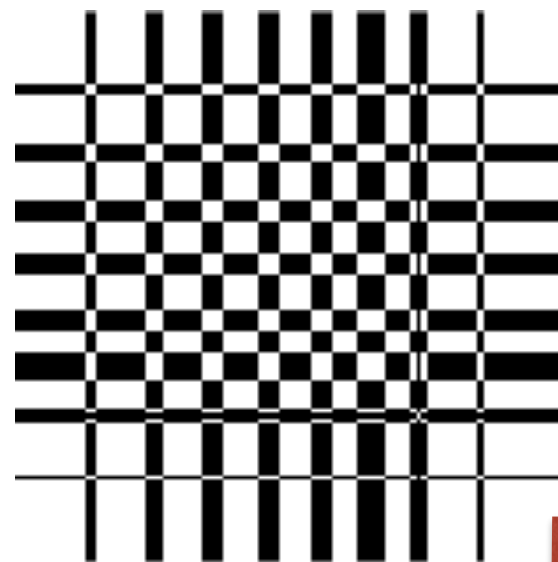
covariance matrices



Gaussian priors on patches

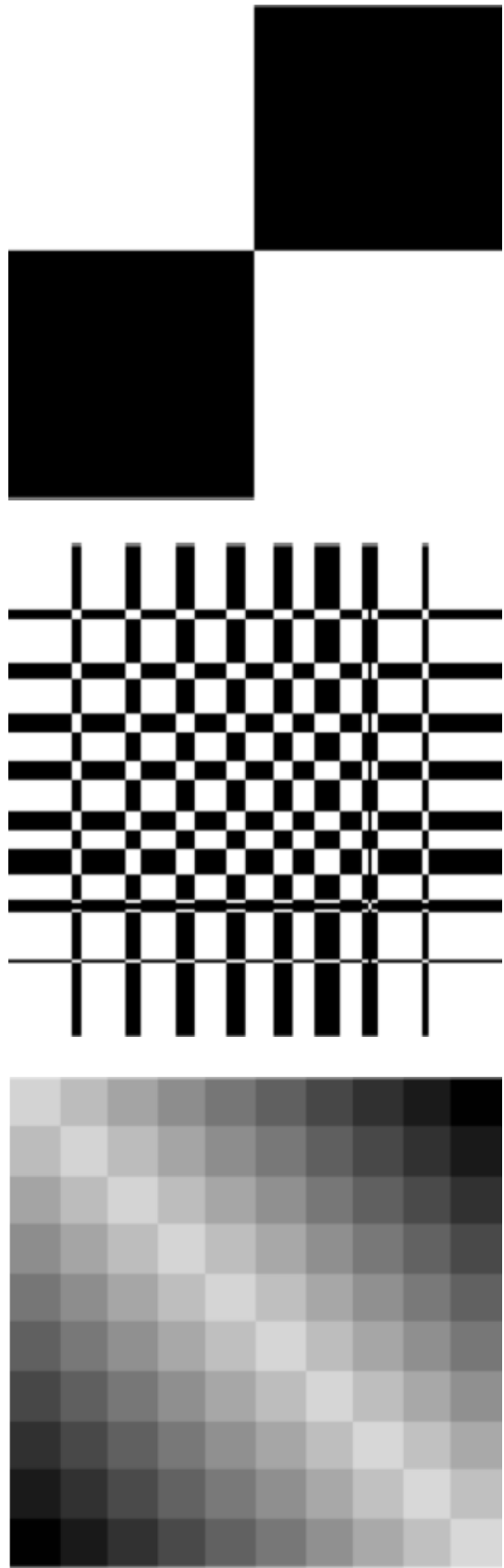


covariance matrices



Gaussian models for denoising

Covariance matrices



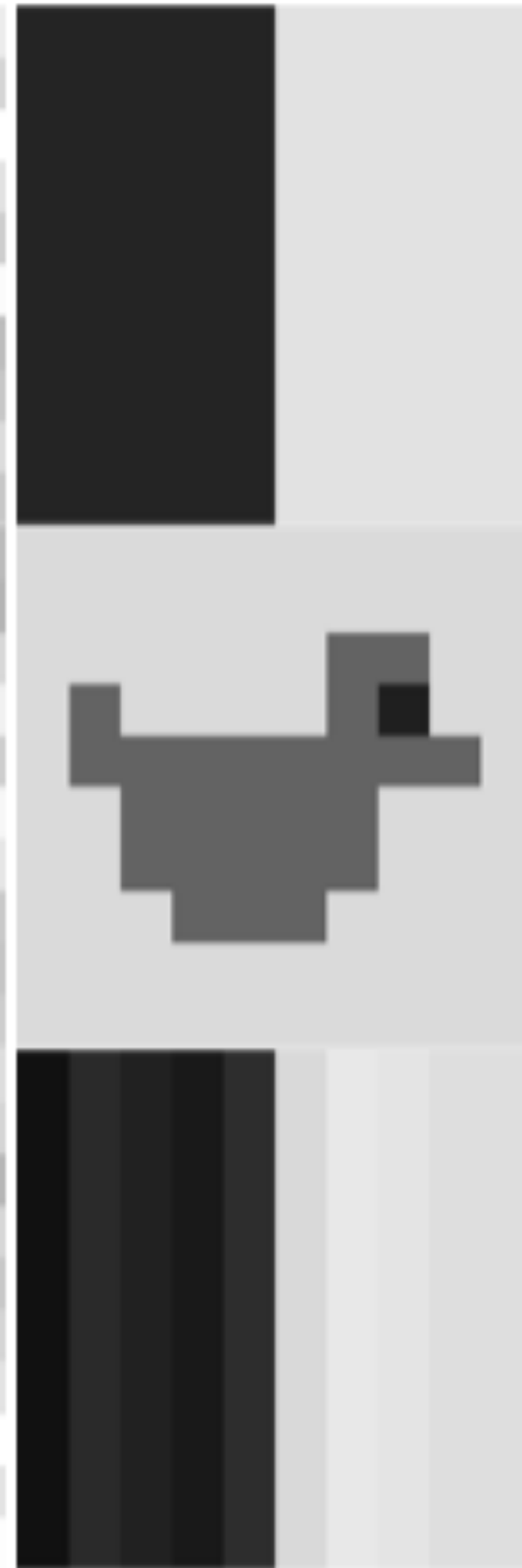
Original patch



Noisy patch



Denoised patch



**How to estimate these
Gaussian or GMM
models?**

Inference of the models?

- Gaussian models: NL-Bayes, Lebrun et al. [13]
- GMM $\sum_{k=1}^K \pi_k \mathcal{N}(\mu_k, \Sigma_k)$
 - ❖ large external database, EPLL, Zoran-Weiss [11]
 - ❖ directly on the degraded image, PLE, Yu et al. [12] ; Teodoro et al. [15], SURE-PLE, Wang et Morel [13], Houdard et al. [18]
- GGMM (generalized Gaussian mixture models) Deledalle et al. [18]

Curse of dimensionality



- Estimation of Sample Covariance Matrices Σ from n samples in high dimension is difficult : estimates tend to be **ill-conditioned** or even singular...
- but Σ has to be inverted to compute estimators...



- use small patches NL-Bayes, Lebrun et al. [13],
- Gaussian of lower dimensions, SURE-PLE, Wang et al. [13] HDMI, Houdard et al [18]
- regularization $(+|\epsilon|I_p)$ or hyperpriors HBE, Aguerrebere et al. [17]

Progress in image denoising



Original image

Progress in image denoising



Noisy image with $\sigma = 50$

Progress in image denoising



Wavelet denoising

Progress in image denoising



NL-Bayes denoising

Progress in image denoising



HDMI denoising

NL-Bayes in Dxo Photolab



Try by yourself ! demo.ipol.im/demo/16/

Patch-based Video inpainting



[Newson et al 2014]

Patch-based Video inpainting



[Newson et al 2014]

Patch-based Video inpainting



[Newson et al 2014]

Patch-based Video inpainting



[Newson et al 2014]

Patch-based Video inpainting



[Newson et al 2014]

Patch-based Video inpainting



[Newson et al 2014]

Multi-image restoration

Multi-image restoration



[technorms.com]

- Denoising
- Low light imaging
- HDR
- Superresolution
- Best image selection...



Olympus



Apple



Google, Night Sight

Multi-image restoration

2009

A NOTE ON MULTI-IMAGE DENOISING

Toni Buades,^{1,4} Yifei Lou², J.M. Morel^{3,1} and Zhongwei Tang³

¹ Universitat de les Illes Balears, Departament de Matemàtiques e Informàtica, Spain

² Department of Mathematics, University of California Los Angeles, USA

³ Centre de Mathématiques et de Leurs Applications, Ecole Normale Supérieure de Cachan, France

⁴ MAP5, Université René Descartes, France.

Deep Burst Denoising

2017

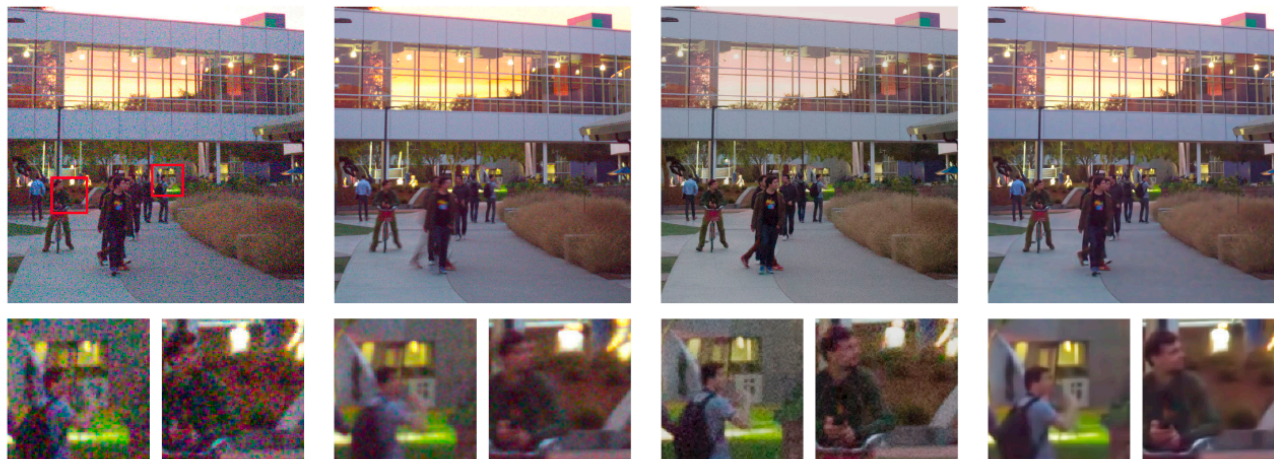
Clément Godard^{1,*}

Kevin Matzen²

Matt Uyttendaele²

¹University College London

²Facebook



Handheld Multi-Frame Super-Resolution

2019

BARTŁOMIEJ WRONSKI, IGNACIO GARCIA-DORADO, MANFRED ERNST, DAMIEN KELLY, MICHAEL KEVIN MIN, CHIA-KAI LIANG, MARC LEVOY, and PEYMAN MILANFAR, Google Research



Fig. 1. We present a multi-frame super-resolution algorithm that supplants the need for demosaicing in a camera pipeline by merging a burst of raw images. We show a comparison to a method that merges frames containing the same-color channels together first, and is then followed by demosaicing (**top**). By contrast, our method (**bottom**) creates the full RGB directly from a burst of raw images. This burst was captured with a hand-held mobile phone and processed on device. Note in the third (red) inset that the demosaiced result exhibits aliasing (Moiré), while our result takes advantage of this aliasing, which changes on every frame in the burst, to produce a merged result in which the aliasing is gone but the cloth texture becomes visible.

Burst photography for high dynamic range and low-light imaging on mobile cameras

Samuel W. Hasinoff
Jonathan T. Barron

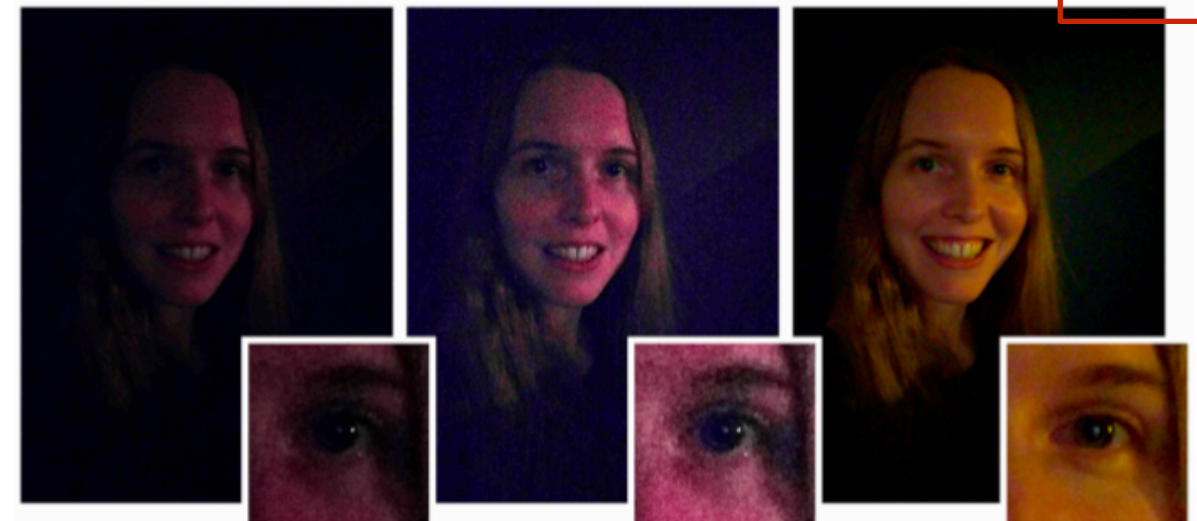
Dillon Sharlet
Florian Kainz

Ryan Geiss
Jiawen Chen

Andrew Adams
Marc Levoy

Google Research

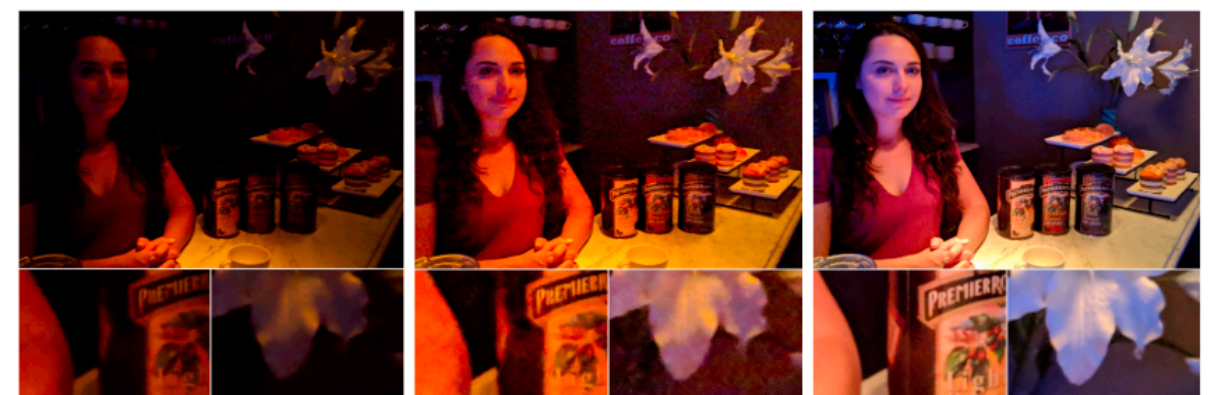
2016



2019

Handheld Mobile Photography in Very Low Light

ORLY LIBA, KIRAN MURTHY, YUN-TA TSAI, TIM BROOKS, TIANFAN XUE, NIKHIL KARNAD, QIURUI HE, JONATHAN T. BARRON, DILLON SHARLET, RYAN GEISS, SAMUEL W. HASINOFF, Yael Pritch, and MARC LEVOY, Google Research



(a) Previously described result

(b) Previously described result, gained

(c) Our result

HDR+, Google



[Hasinoff et al., 2016]

HDR+, Google



[Hasinoff et al., 2016]

- Raw burst alignment and merging, only classical image processing
- Embedded in smartphones (inc. 1st Google Pixel)
- Refined later in Night Sight [Liba et al., 2019]

Photographing paintings by image fusion



[Haro, Buadès, Morel, 2012]

See www.ipol.im for an online demo.

Photographing paintings by image fusion

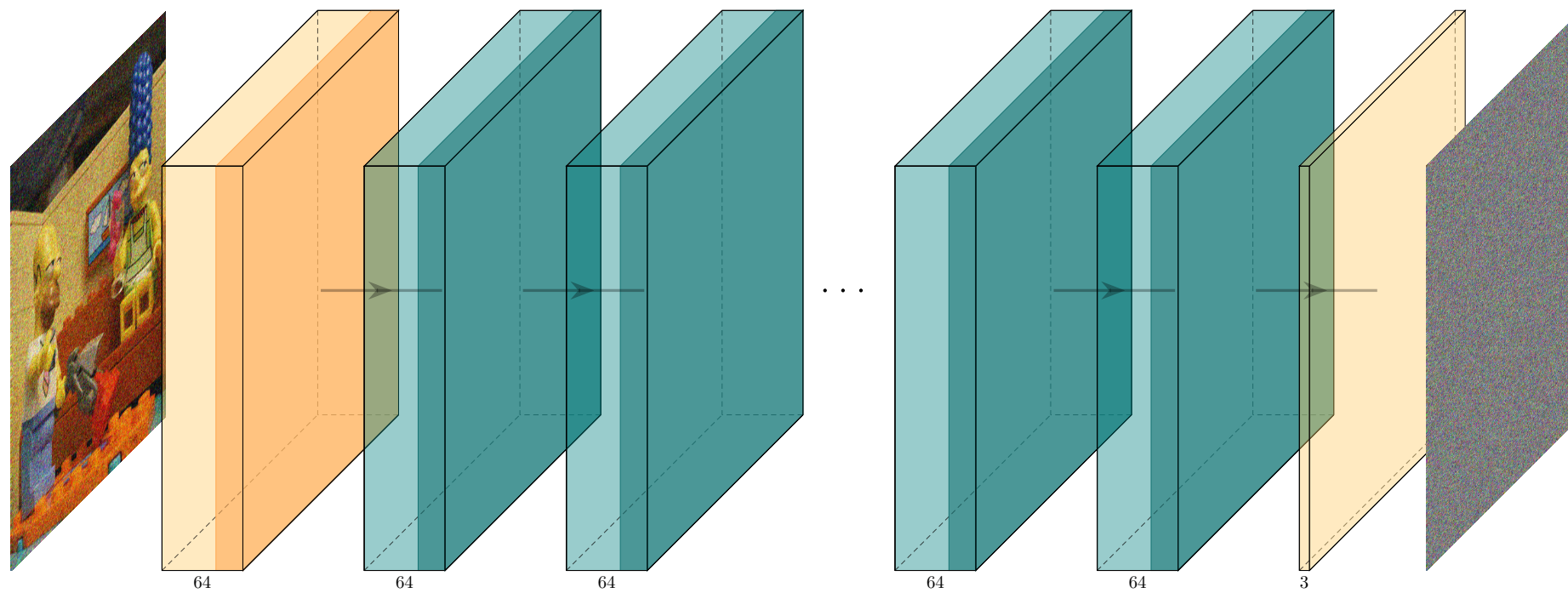


[Haro, Buadès, Morel, 2012]

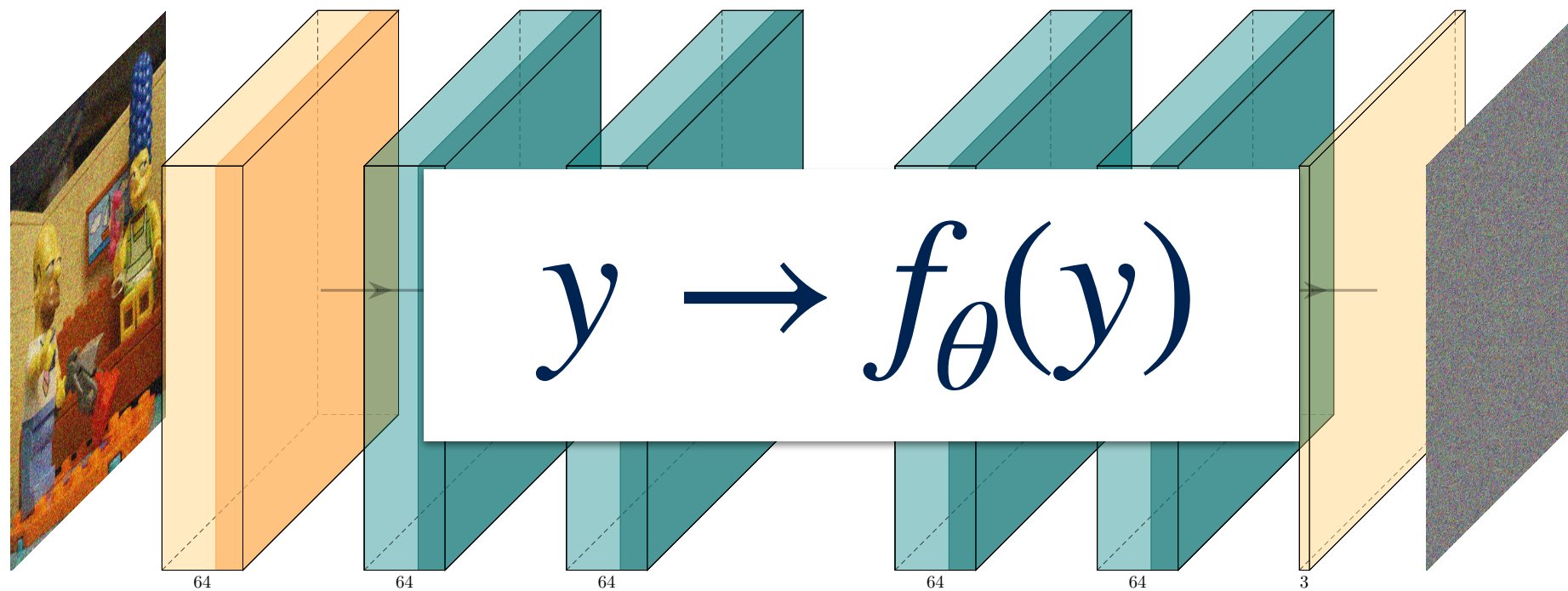
See www.ipol.im for an online demo.

Neural networks and implicit image models

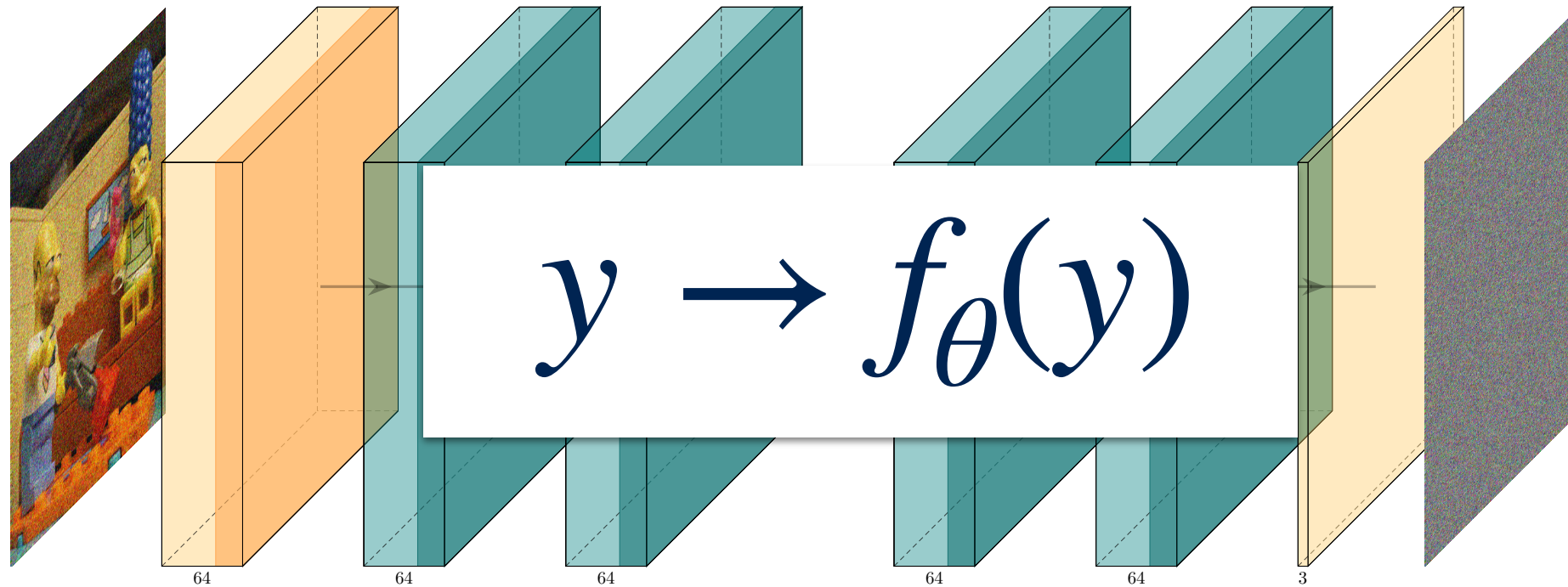
End-to-end neural networks



End-to-end neural networks

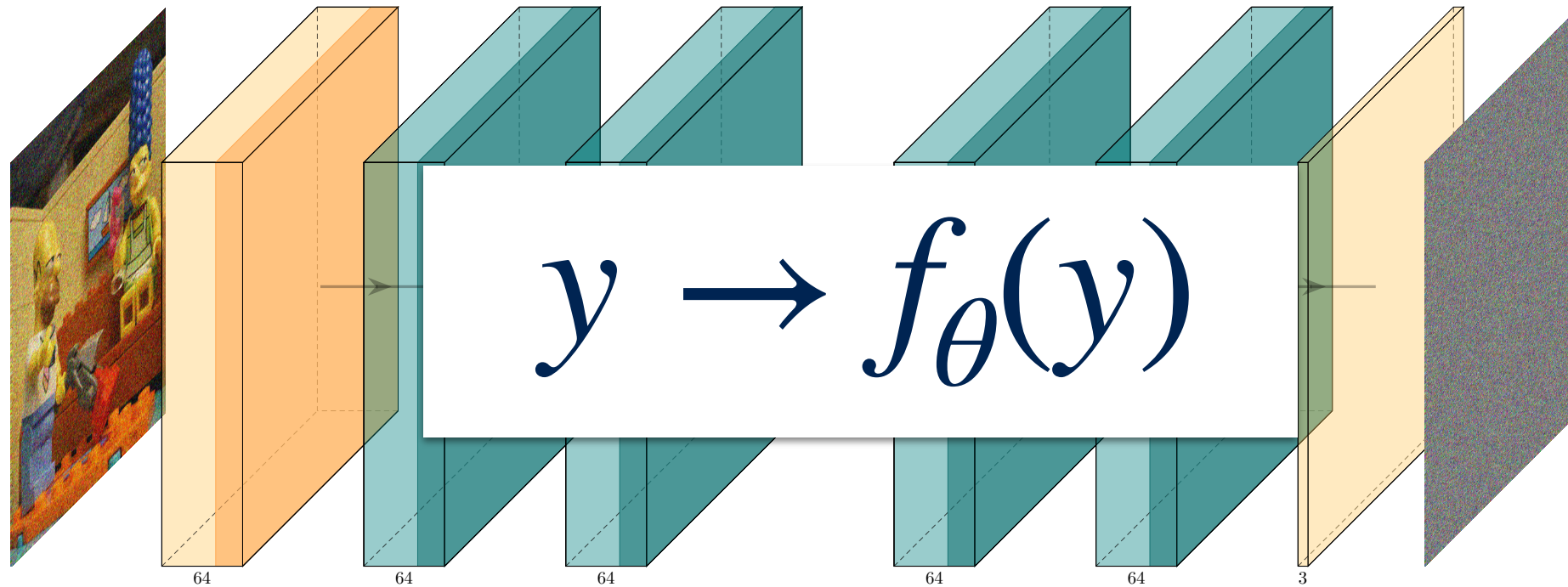


End-to-end neural networks



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

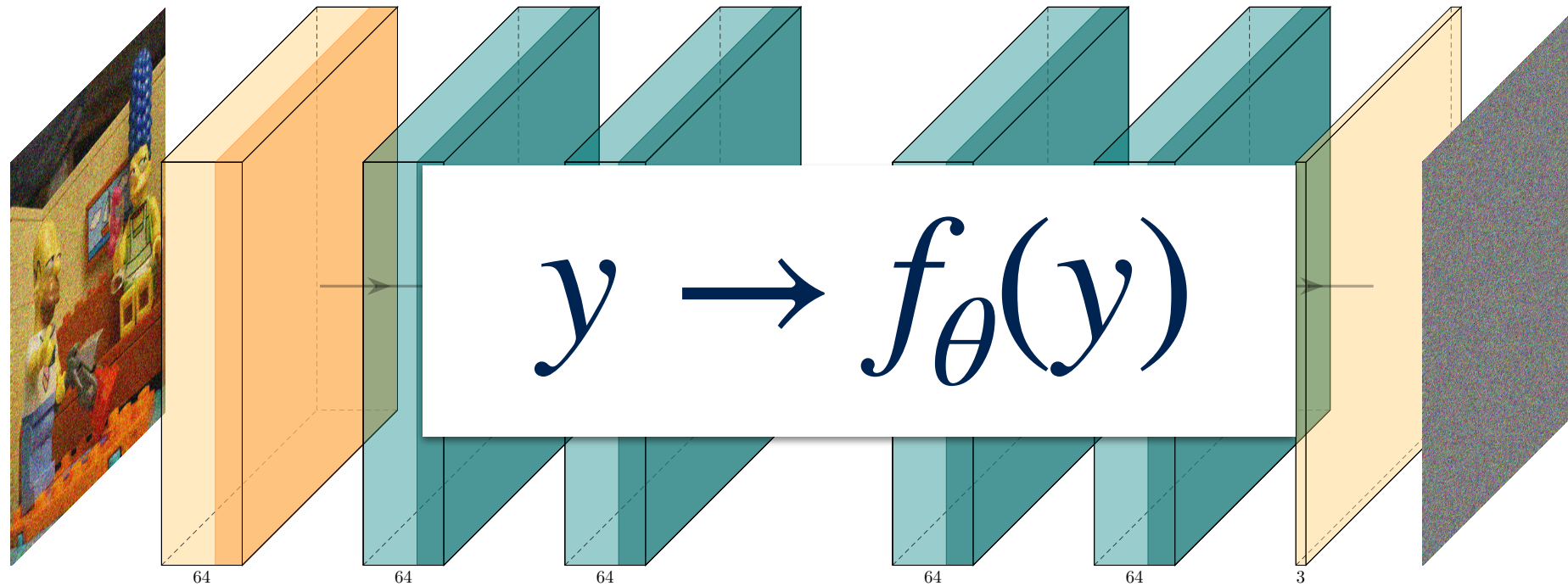
End-to-end neural networks



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

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

End-to-end neural networks

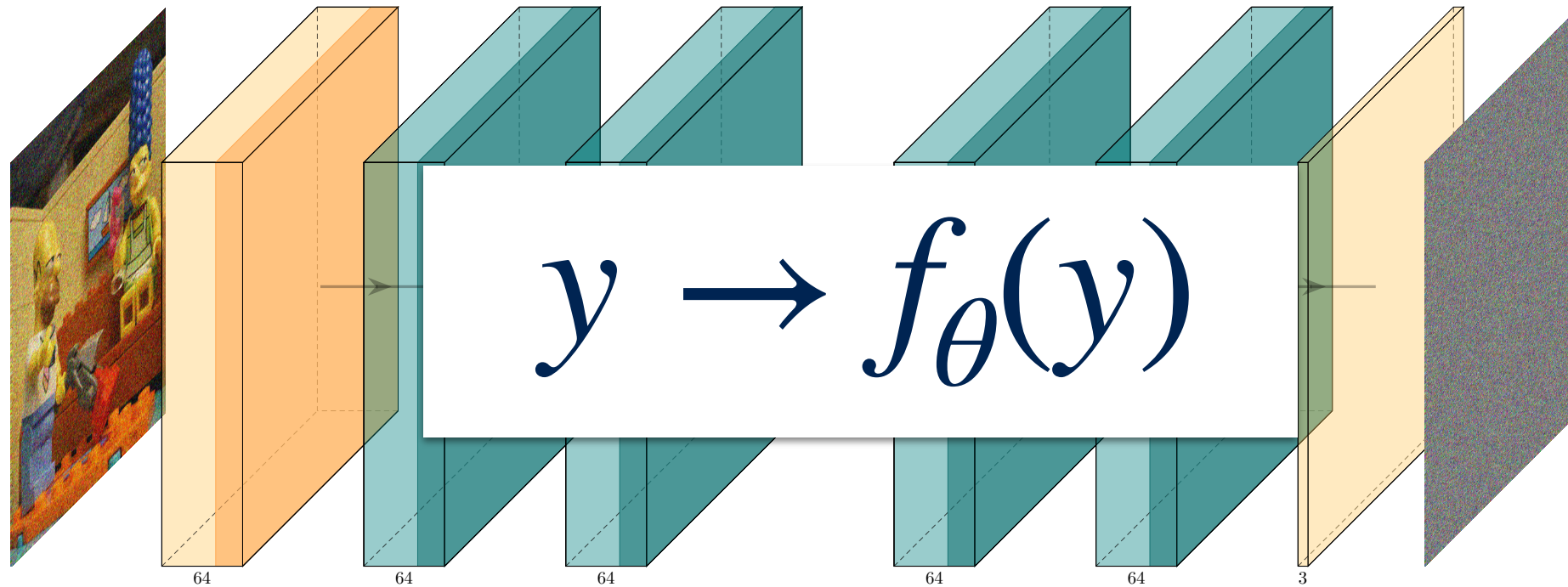


Set $(x_i, y_i)_{i=1}^N$ of independent realizations of the degradation model

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

$$\simeq \mathbb{E}[\mathcal{L}(f_{\theta}(Y) - X)]$$

End-to-end neural networks



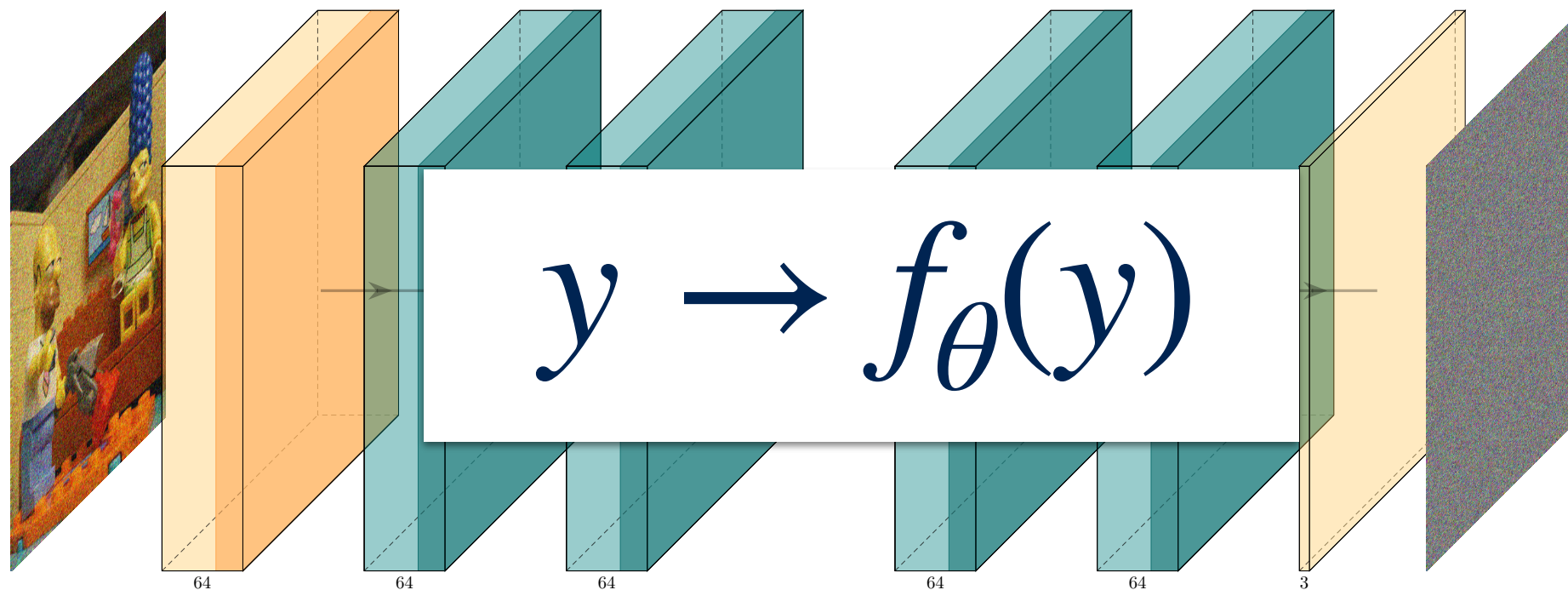
Set $(x_i, y_i)_{i=1}^N$ of independent realizations of the degradation model

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

$$\simeq \mathbb{E}[\mathcal{L}(f_{\theta}(Y) - X)]$$

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

End-to-end neural networks



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

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

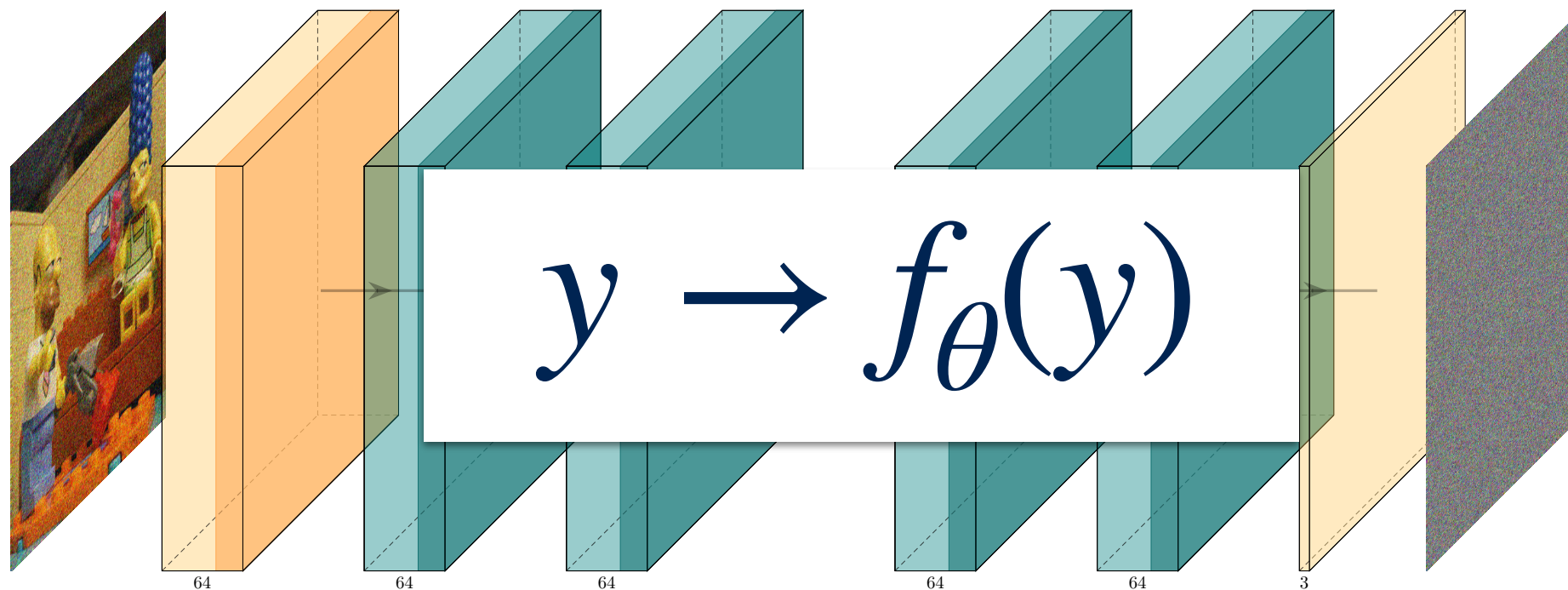
$\simeq \mathbb{E}[\mathcal{L}(f_{\theta}(Y) - X)]$

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

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

- Finite number of samples N
- Limited expressive power f_{θ}
- Optimization not perfect

End-to-end neural networks



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

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

$\simeq \mathbb{E}[\mathcal{L}(f_{\theta}(Y) - X)]$

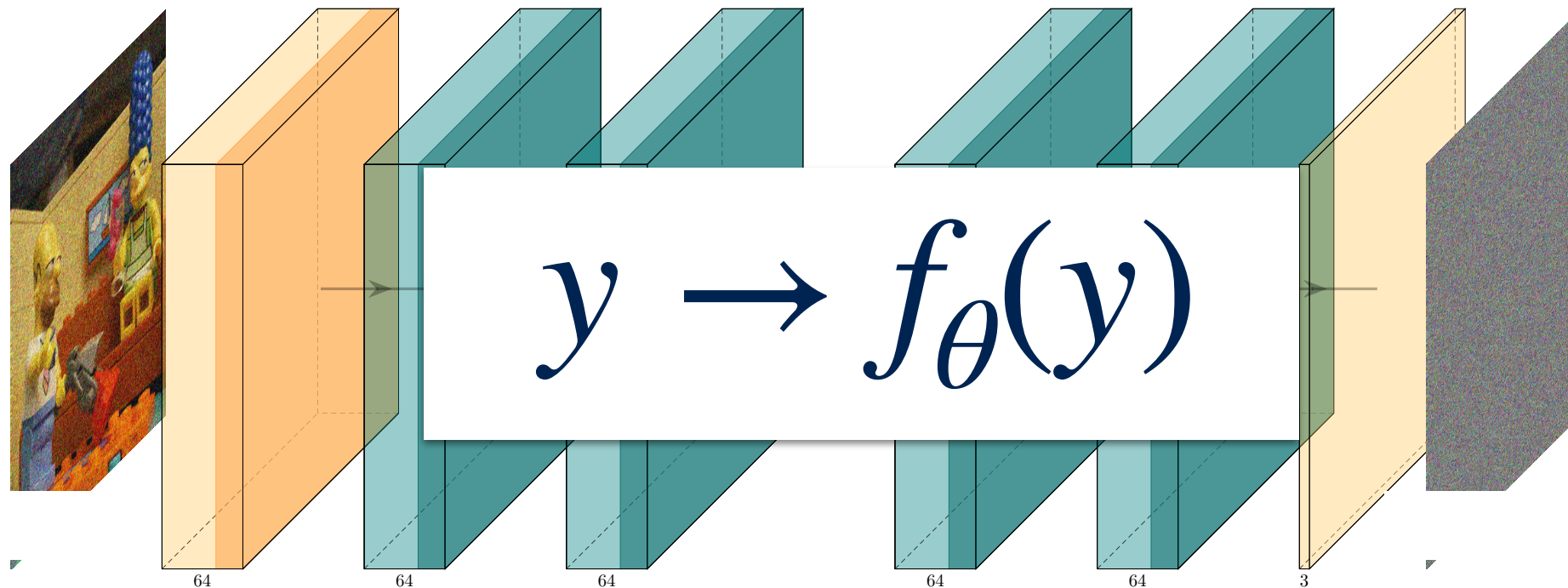
If $\mathcal{L}(v, w) = \|v - w\|^2$
 $f_{\hat{\theta}}(y) \simeq \mathbb{E}[X | Y = y]$

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

End-to-end neural networks



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

Training:
$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{\theta}(y_i), x_i)$$

$$\simeq \mathbb{E}[\mathcal{L}(f_{\theta}(Y) - X)]$$

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

Practical limitations:

1. **Problem specific**, must be trained again for a new problem (even if recent progresses have been made in this direction).
2. Compute **pointwise estimators**. Uncertainty quantification?

Progress in image denoising



Noisy image with $\sigma = 50$

Progress in image denoising



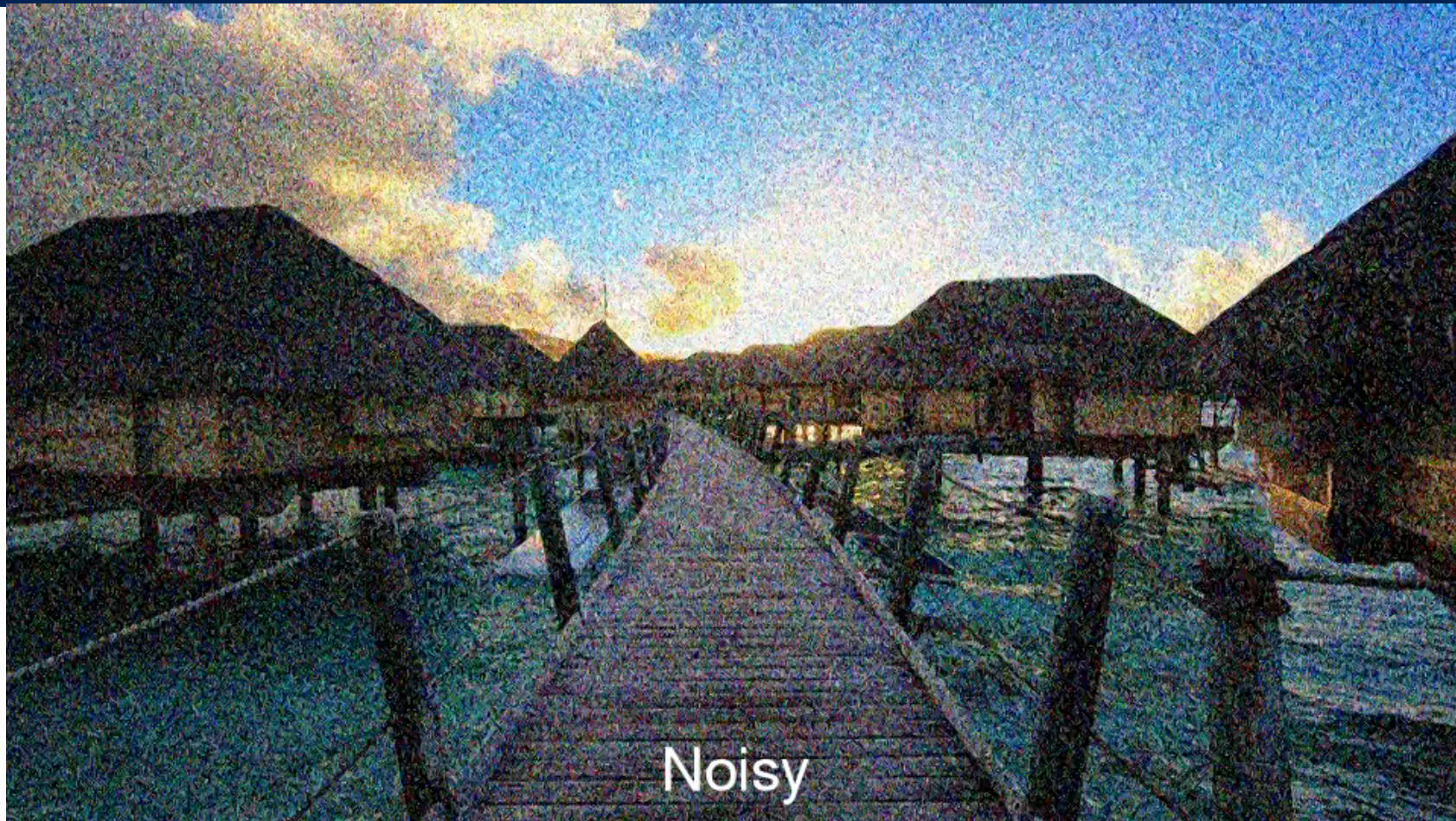
Wavelet denoising

Progress in image denoising



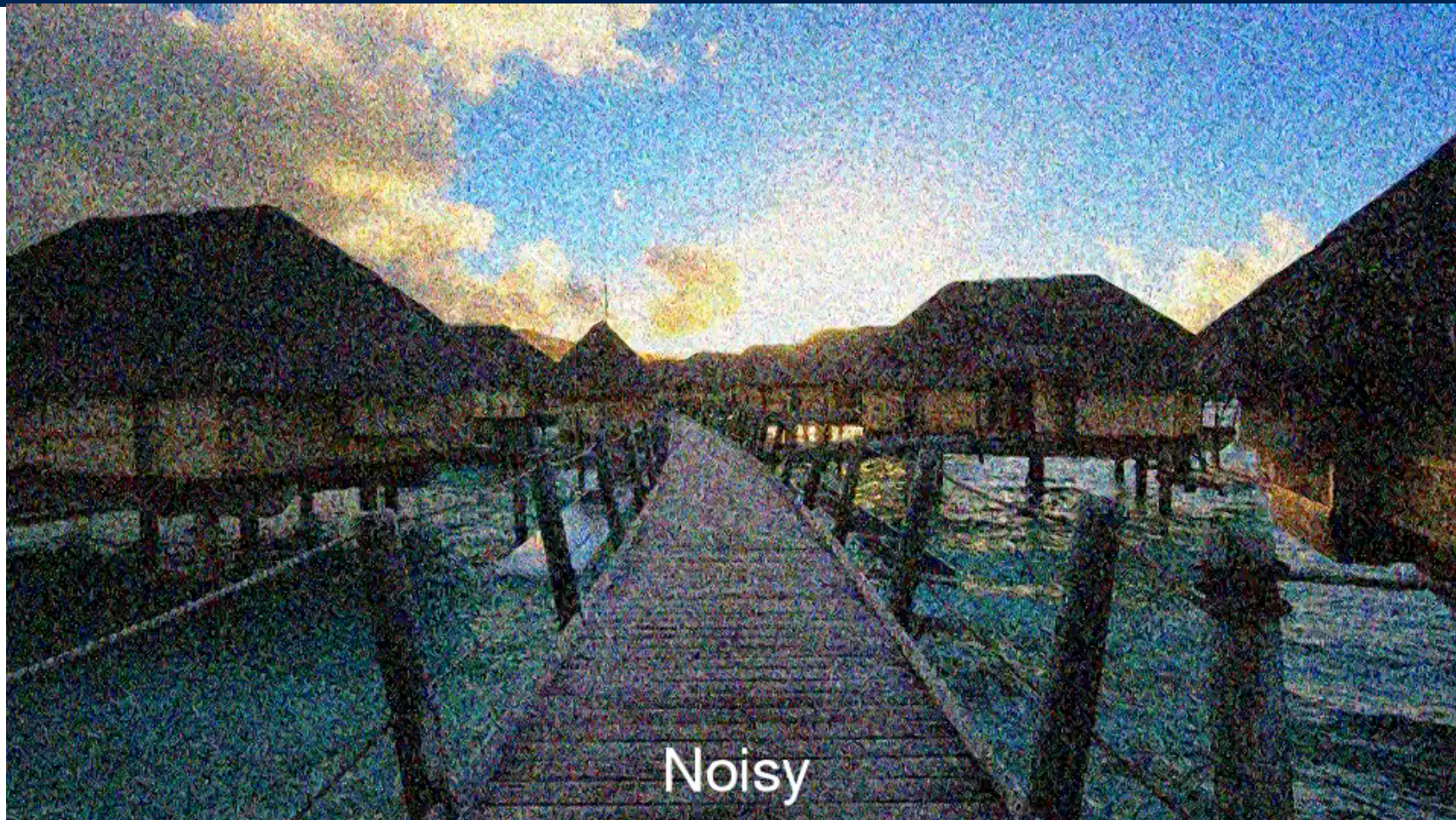
Denoising with FFDnet [Zhang et al. 2018]

Video denoising



[DVDnet, Tassano,
D., Veit, 2019]

Video denoising



[DVDnet, Tassano,
D., Veit, 2019]

Plug-and-Play

$$y = Ax + n$$

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

Plug-and-Play

$$y = Ax + n$$

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

$$\text{prox}_{\gamma U}(y) = \arg \min_x \frac{1}{2\gamma} \|y - x\|_2^2 + U(x)$$

Plug-and-Play

$$y = Ax + n$$

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

$$\text{prox}_{\gamma U}(y) = \arg \min_x \frac{1}{2\gamma} \|y - x\|_2^2 + U(x)$$

$\text{prox}_{\gamma U}$ solves a denoising problem for the regularizer U !

$$y = x + n \quad \text{with} \quad n \sim \mathcal{N}(0, \sqrt{\gamma}).$$

Plug-and-Play

$$y = Ax + n$$

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

$$\text{prox}_{\gamma U}(y) = \arg \min_x \frac{1}{2\gamma} \|y - x\|_2^2 + U(x)$$

$\text{prox}_{\gamma U}$ solves a denoising problem for the regularizer U !

$$y = x + n \quad \text{with} \quad n \sim \mathcal{N}(0, \sqrt{\gamma}).$$

Plug and Play: replace $\text{prox}_{\gamma U}$ in proximal optimization schemes by a well chosen denoiser D_γ .

Plug-and-Play

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

Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

- **Versatility/flexibility:** one network to rule them all!



Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

- **Versatility/flexibility:** one network to rule them all!
- Used with $\neq$ optimization schemes and $\neq$ denoisers [Meinhardt et al. 2017, Ryu et al. 2019, Xu et al. 2020, Sun et al. 2021,...]



Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

- **Versatility/flexibility:** one network to rule them all!
- Used with $\neq$ optimization schemes and $\neq$ denoisers [Meinhardt et al. 2017, Ryu et al. 2019, Xu et al. 2020, Sun et al. 2021,...]
- State-of the-art results when the denoiser is powerful enough [DPIR, Zhang et al. 2021]



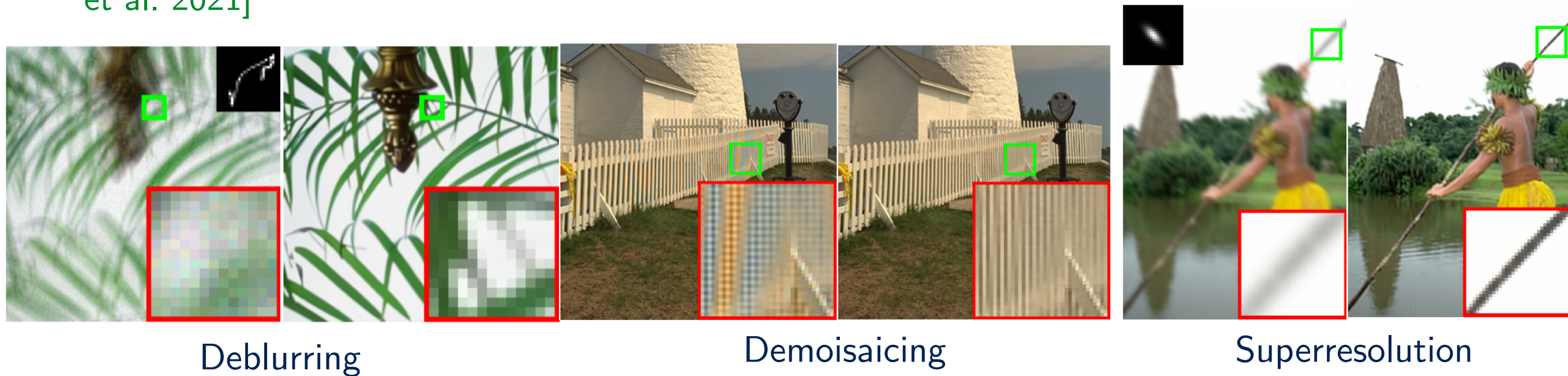
Plug-and-Play

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

Proximal gradient (Forward-Backward splitting)

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

- **Versatility/flexibility:** one network to rule them all!
- Used with $\neq$ optimization schemes and $\neq$ denoisers [Meinhardt et al. 2017, Ryu et al. 2019, Xu et al. 2020, Sun et al. 2021,...]
- State-of-the-art results when the denoiser is powerful enough [DPIR, Zhang et al. 2021]



Plug-and-Play

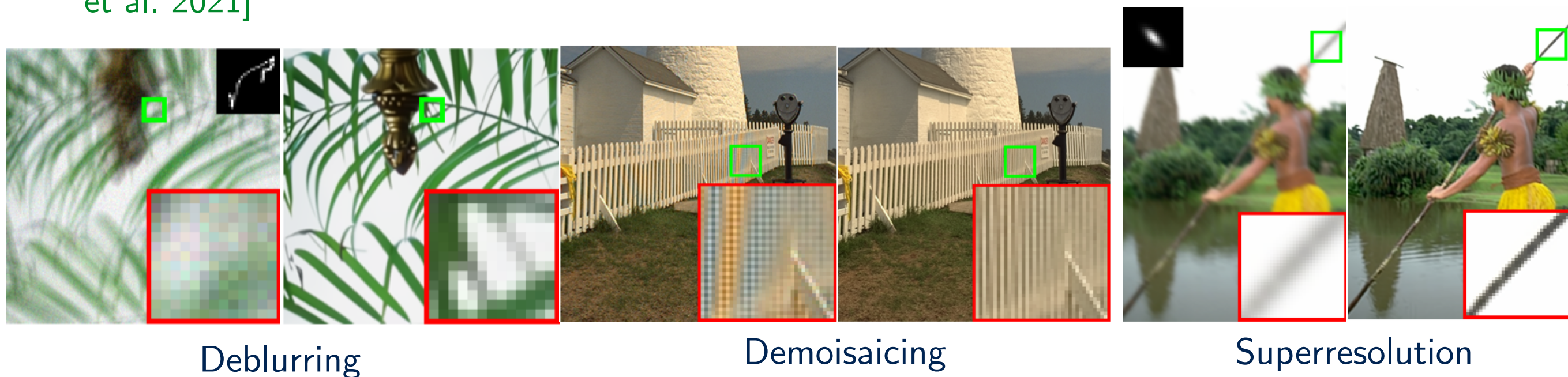
$$\operatorname{argmin}_x F(x, y) + U(x)$$

Proximal gradient (Forward-Backward splitting)

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



- **Versatility/flexibility:** one network to rule them all!
- Used with $\neq$ optimization schemes and $\neq$ denoisers [Meinhardt et al. 2017, Ryu et al. 2019, Xu et al. 2020, Sun et al. 2021,...]
- State-of-the-art results when the denoiser is powerful enough [DPIR, Zhang et al. 2021]



Convergence analysis under appropriate conditions on F and U

PnP video - deblurring



aerobatics blurred 22.84dB



PnP-V DRUNet (eps=20 alpha=1.25) 37.94dB



DPIR 38.83dB



PnP-V FastDVDnet (eps=20 alpha=1.0) 38.37dB

- Solution for embeddable systems with limited memory [Monod et al. 2022]

PnP video - deblurring



aerobatics blurred 22.84dB



PnP-V DRUNet (eps=20 alpha=1.25) 37.94dB



DPIR 38.83dB



PnP-V FastDVDnet (eps=20 alpha=1.0) 38.37dB

- Solution for embeddable systems with limited memory [Monod et al. 2022]

PnP video - interpolation



aerobatics clean



aerobatics masked ($\rho=0.9$) 6.17dB



PnP-V DRUNet ($\epsilon=50$ $\alpha=2.75$) 25.16dB



PnP-V FastDVDnet ($\epsilon=40$ $\alpha=1.5$) 33.37dB

- Solution for embeddable systems with limited memory [Monod et al. 2022]

PnP video - interpolation



aerobatics clean



aerobatics masked ($\rho=0.9$) 6.17dB



PnP-V DRUNet ($\epsilon=50$ $\alpha=2.75$) 25.16dB



PnP-V FastDVDnet ($\epsilon=40$ $\alpha=1.5$) 33.37dB

- Solution for embeddable systems with limited memory [Monod et al. 2022]

What about bayesian sampling?

Plug-and-Play sampling methods

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

- Use MCMC / Langevin algorithms with a denoiser to approximate the score $\nabla \log p$
- Strong link with **score matching** and **diffusion models** for image generation



**Cours generative models for
imaging au 2nd semestre**



→ Idea appears in [Alain and Bengio 2014]. Similar ideas in [Guo et al. 2019], [Kadkhodaie et al. 2020], [Kawar et al. 2020], first proof of convergence for PnP-Langevin in [Laumont et al. 2021]

Plug-and-Play sampling methods

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

- Use MCMC / Langevin algorithms with a denoiser to approximate the score $\nabla \log p$
- Strong link with **score matching** and **diffusion models** for image generation



**Cours generative models for
imaging au 2nd semestre**



→ Idea appears in [Alain and Bengio 2014]. Similar ideas in [Guo et al. 2019], [Kadkhodaie et al. 2020], [Kawar et al. 2020], first proof of convergence for PnP-Langevin in [Laumont et al. 2021]

Sampling $p_{X|Y}$ with convergence guarantees

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

Sampling $p_{X|Y}$ with convergence guarantees

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

→ **Ergodic theorem:** if (X_n) has a unique stationary prob. measure $p_{X|Y}$, then

$$\forall f \in L^1(p_{X|Y}), \quad \frac{1}{n} \sum_{k=1}^n f(X_k) \xrightarrow[n \rightarrow \infty]{} \int f(x) dp_{X|Y}(x) \text{ a.s.}$$

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

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

Sampling $p_{X|Y}$ with convergence guarantees

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

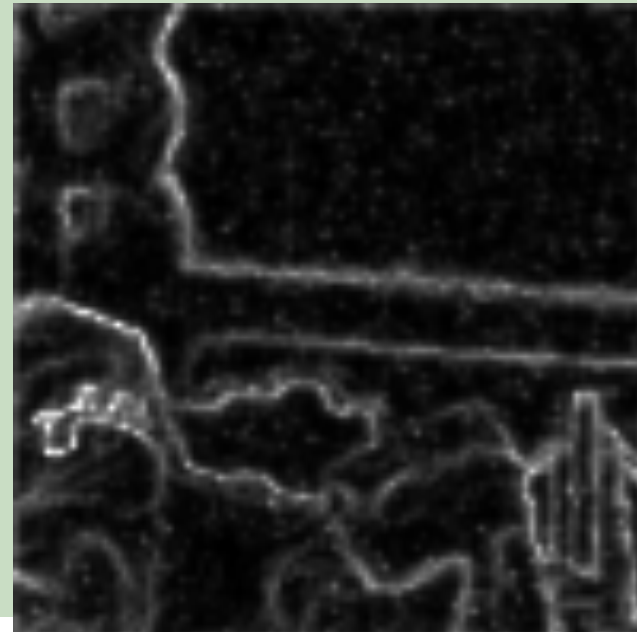
→ **Ergodic theorem:** if (X_n) has a unique stationary prob. measure $p_{X|Y}$, then

$$\forall f \in L^1(p_{X|Y}), \quad \frac{1}{n} \sum_{k=1}^n f(X_k) \xrightarrow[n \rightarrow \infty]{} \int f(x) dp_{X|Y}(x) \text{ a.s.}$$

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

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

TV-LSE: Metropolis for $p_{X|Y}(x) \propto e^{-\lambda TV(x) - \|x-y\|^2/2\sigma^2}$ [Louchet et al. 2008]



Ok for
A working
locally.

Observation y ($\sigma = 0.05$) TVL2 MMSE (psnr = 31.56) TVL2 std

Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$

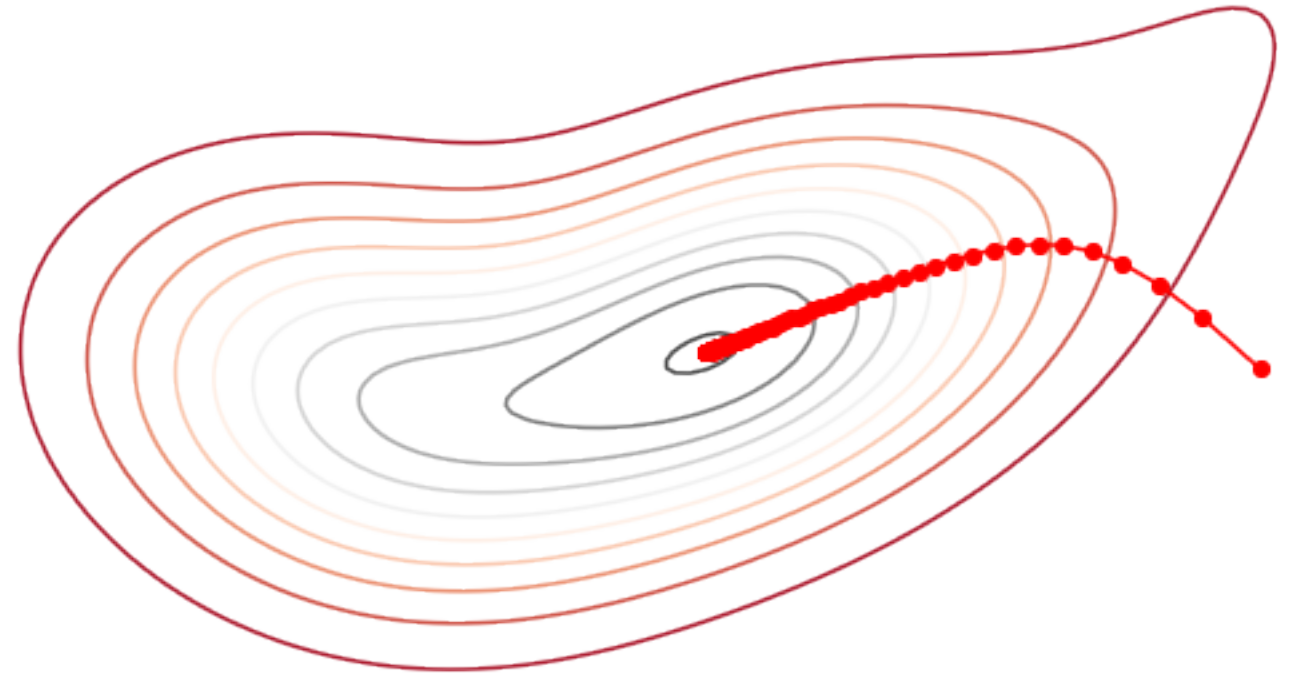
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



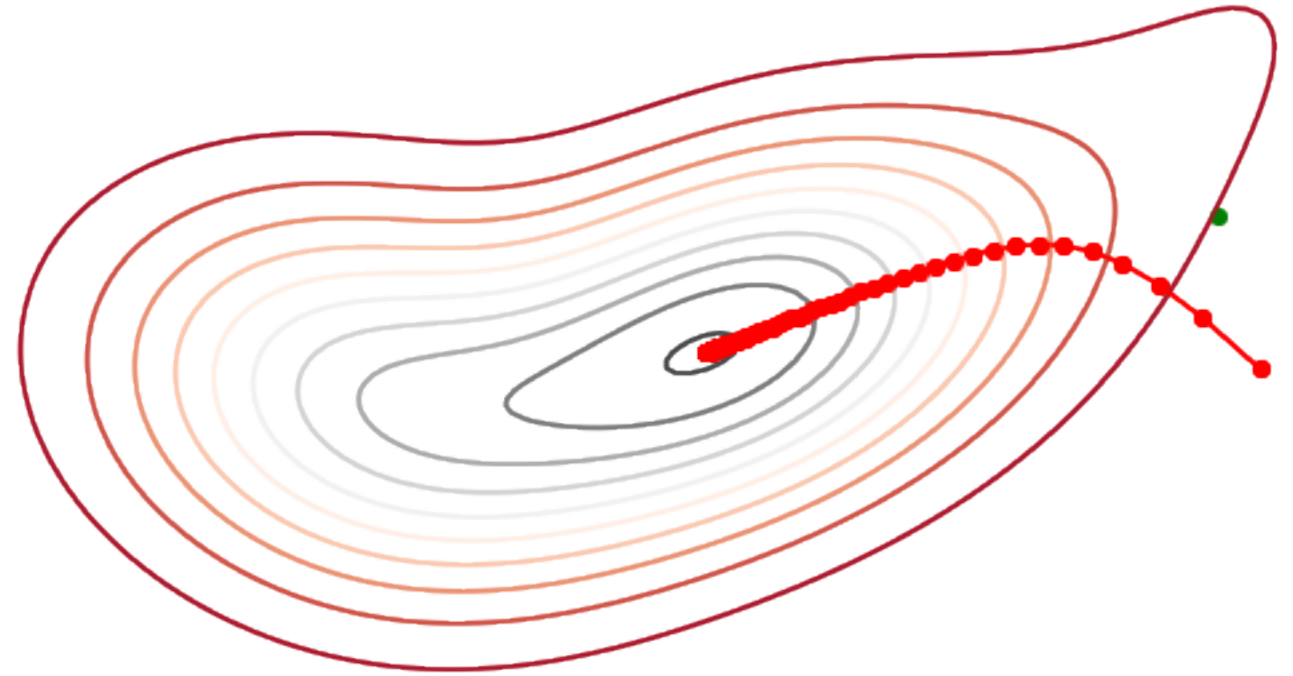
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



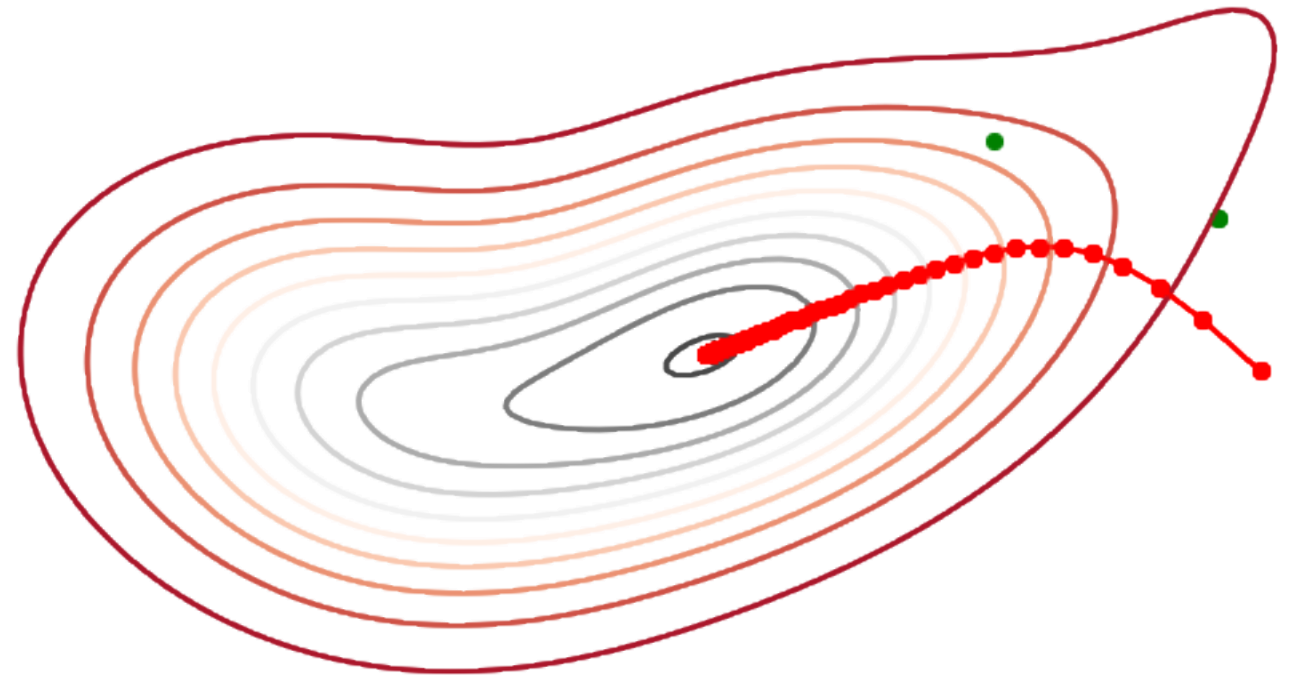
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



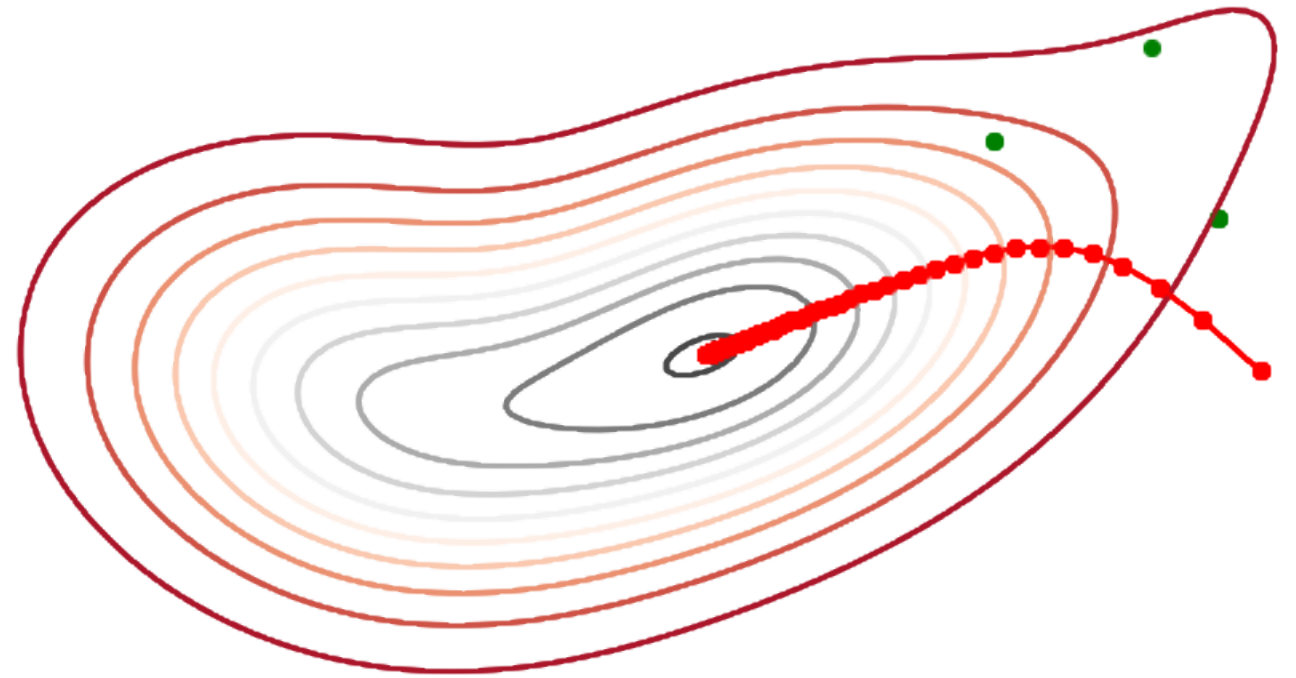
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



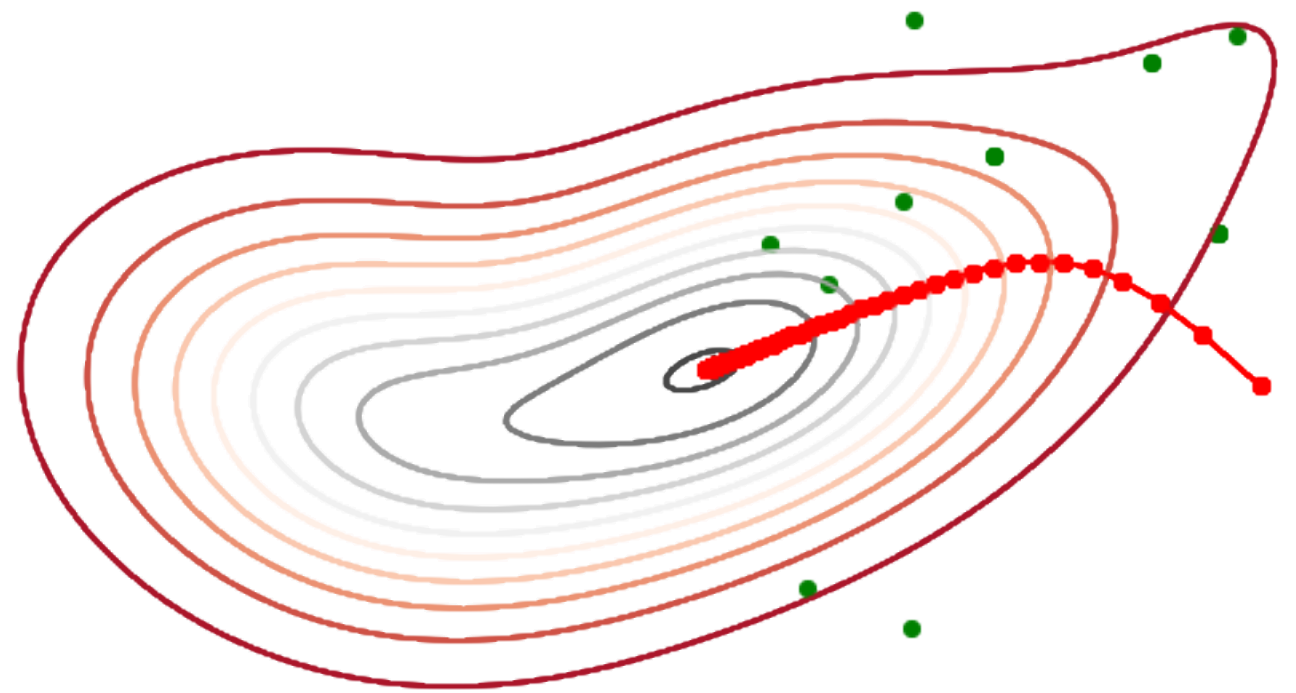
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



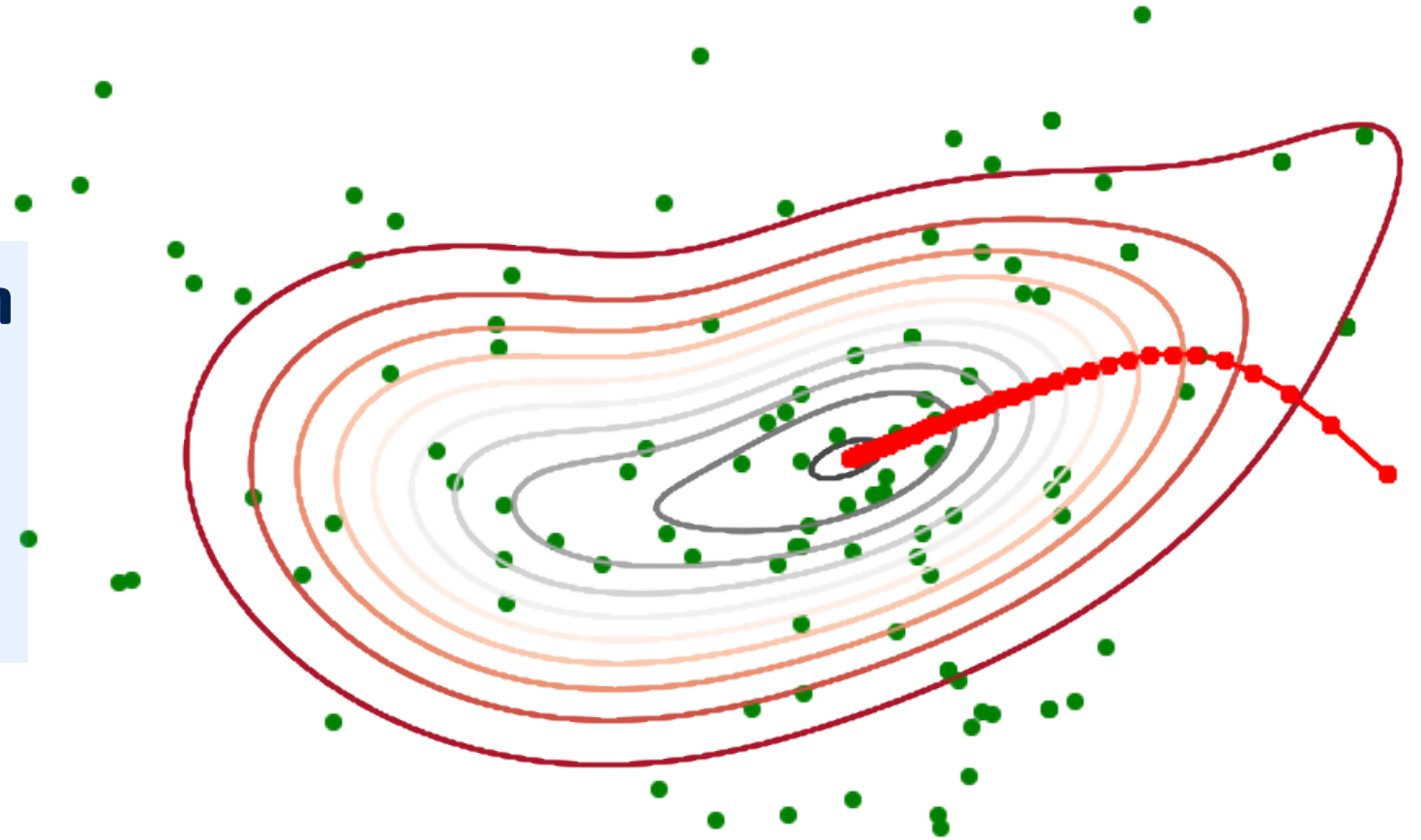
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



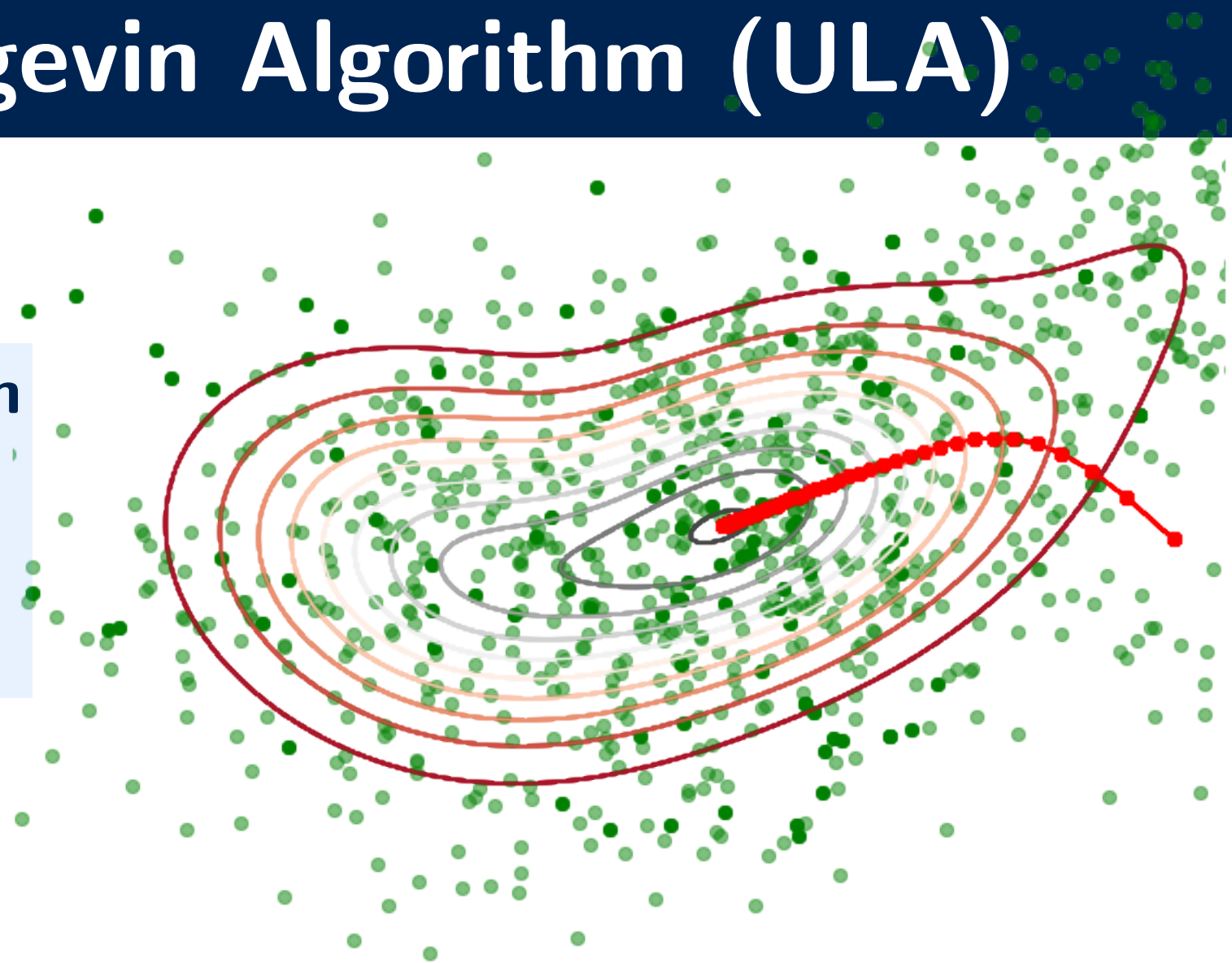
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



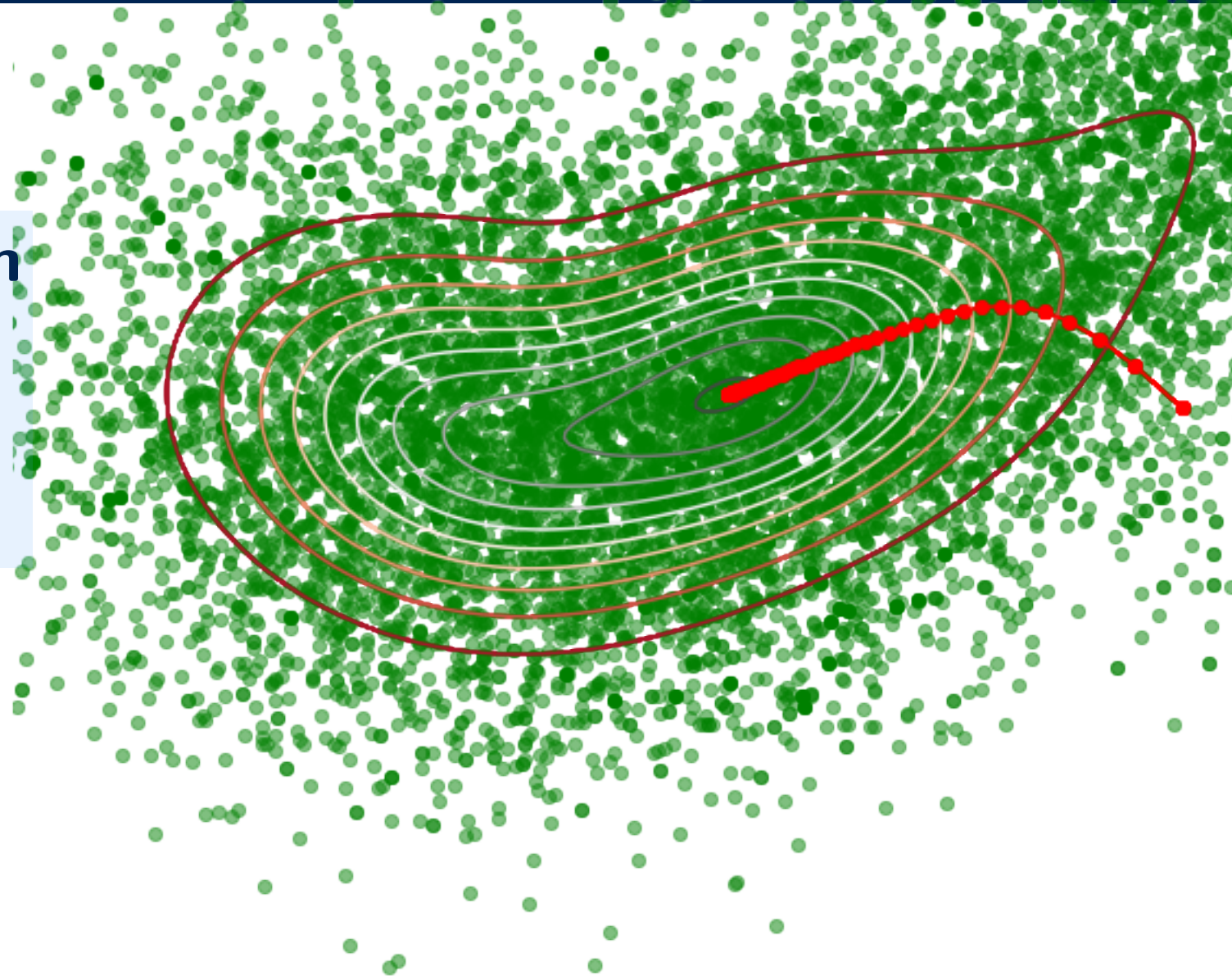
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



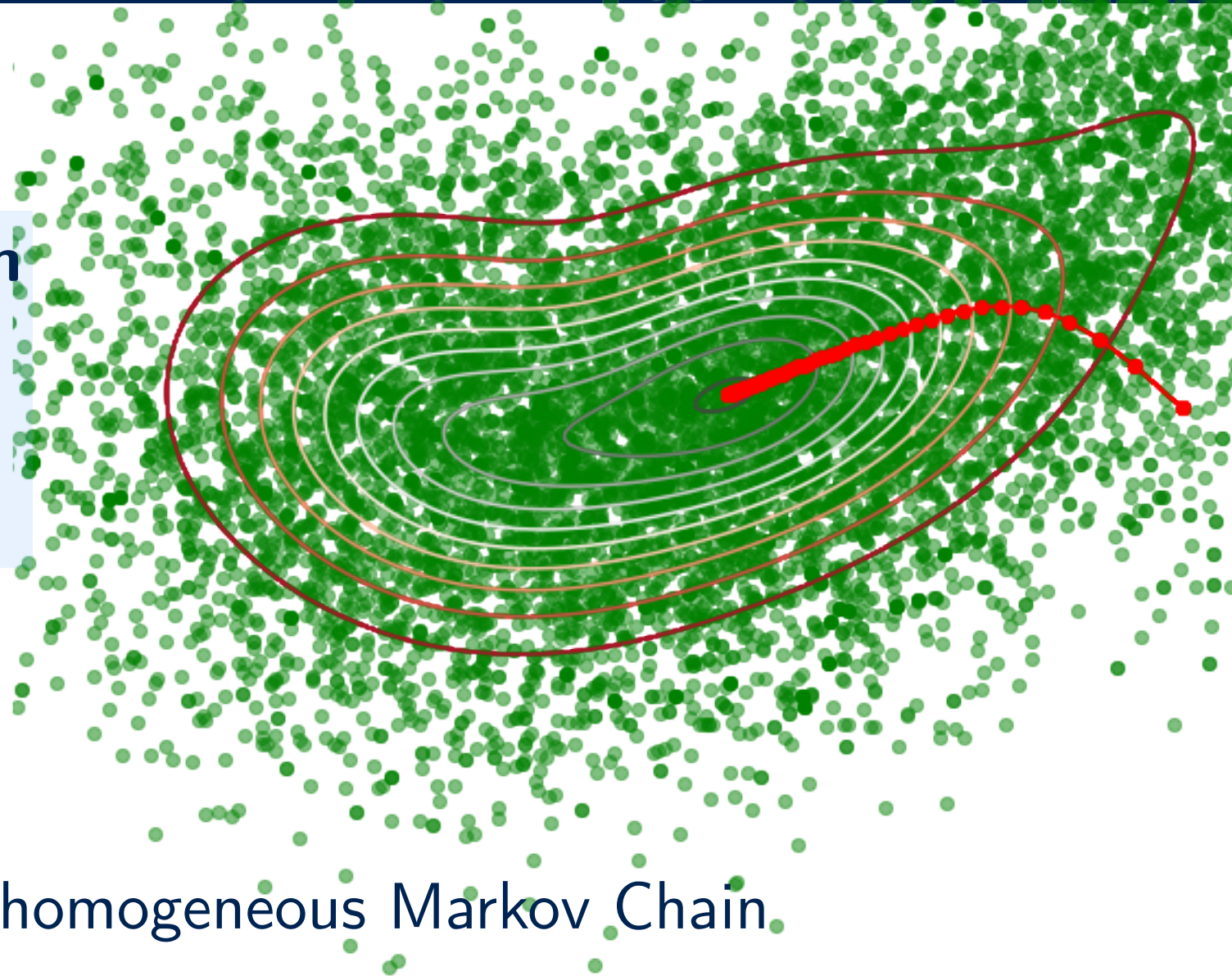
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.

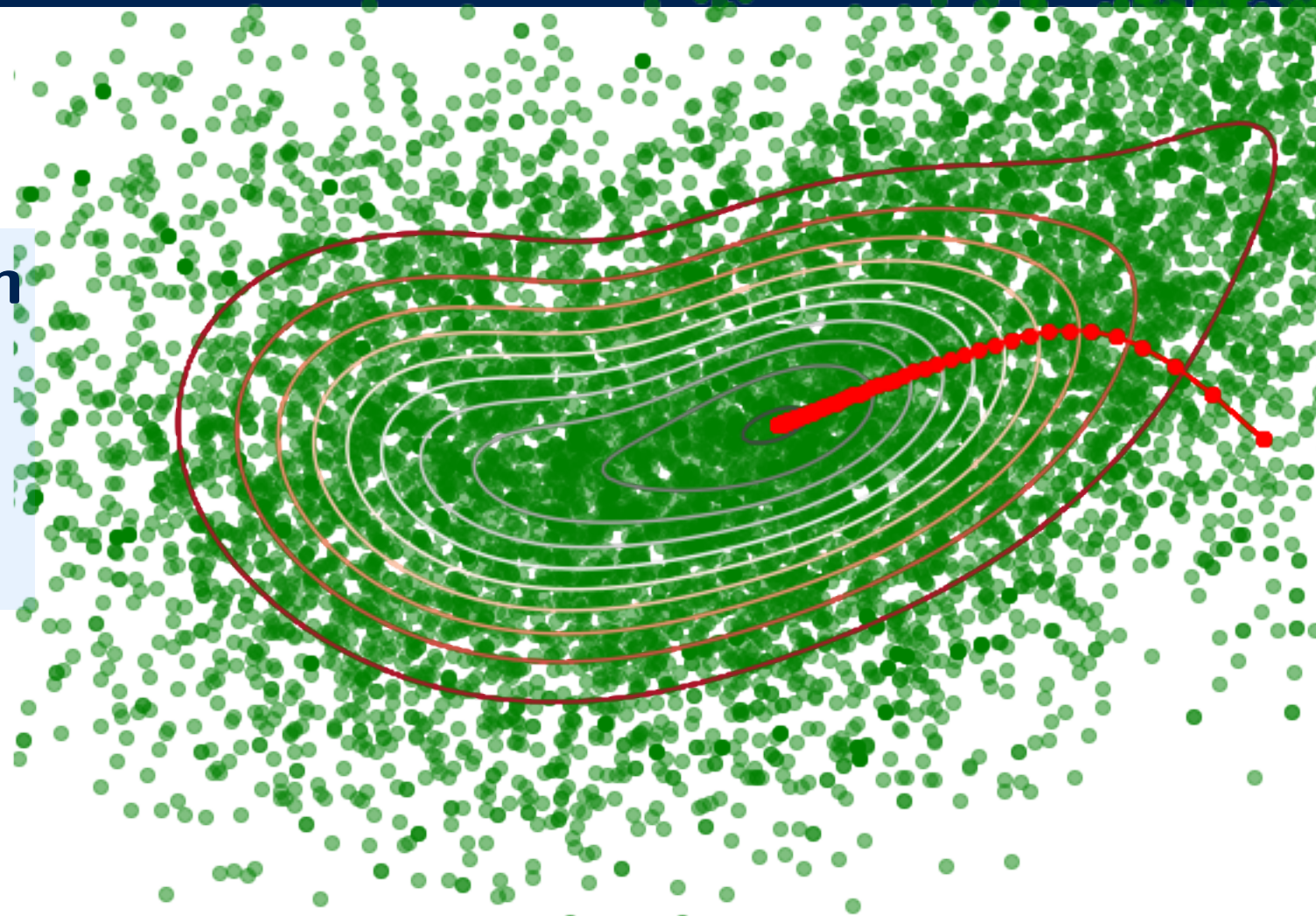
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.
- Euler-Maruyama discretization of the Langevin eq.

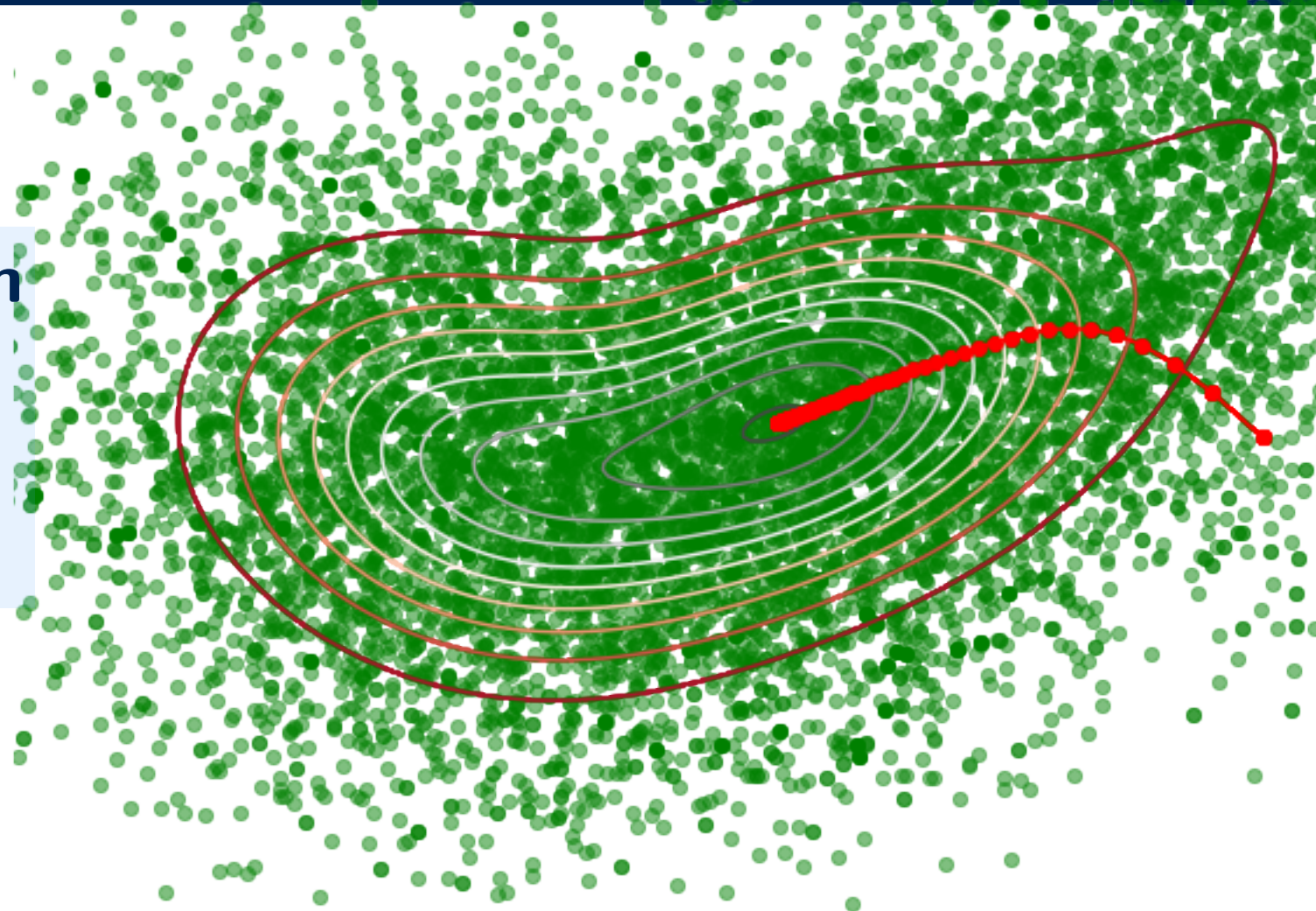
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.
- Euler-Maruyama discretization of the Langevin eq.
- **Convergence** to unique stationary dist. $\pi_\delta \neq \pi$ if ∇H is L -Lipschitz and $\delta < 1/L$

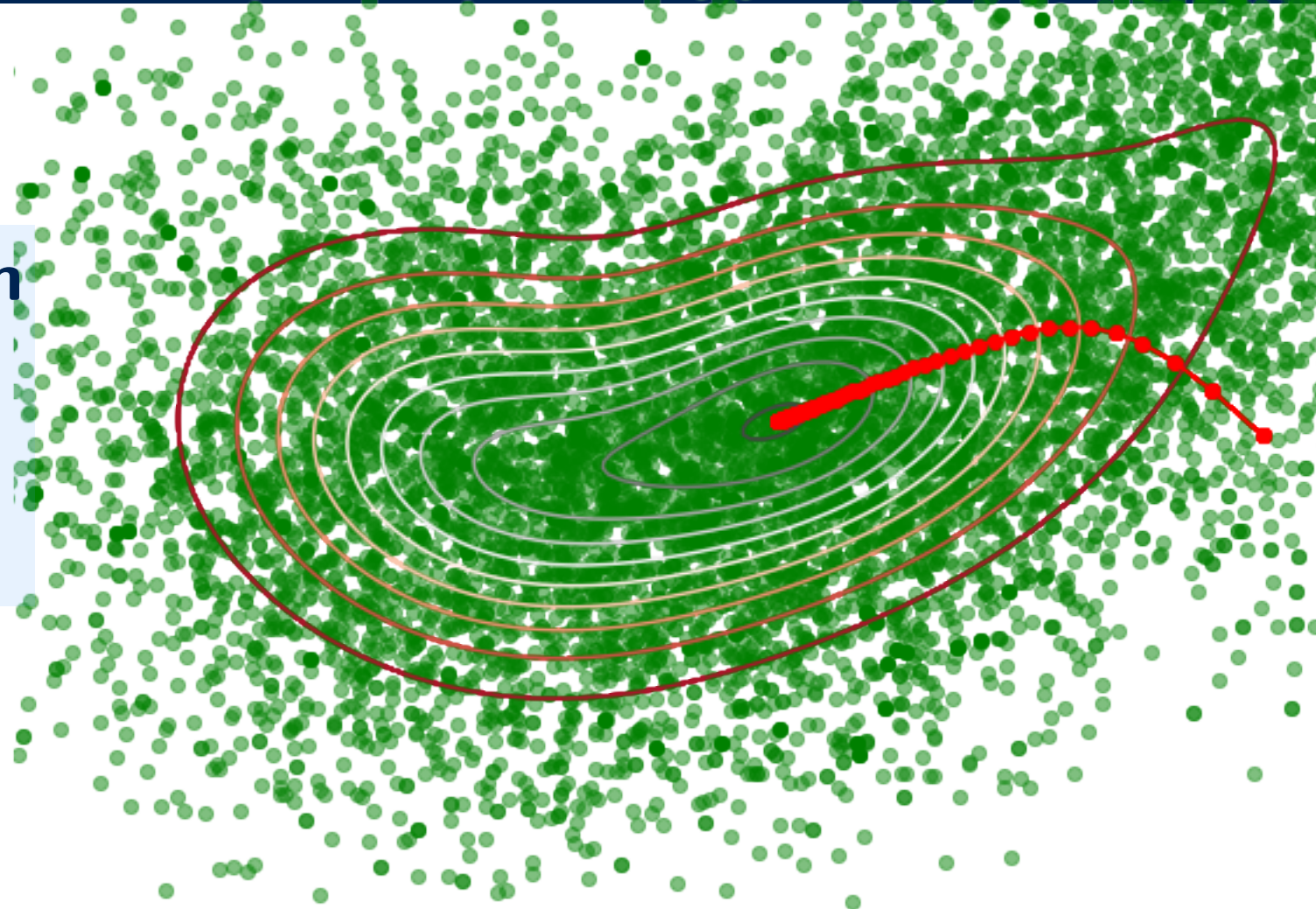
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.
- Euler-Maruyama discretization of the Langevin eq.
- **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 ∞

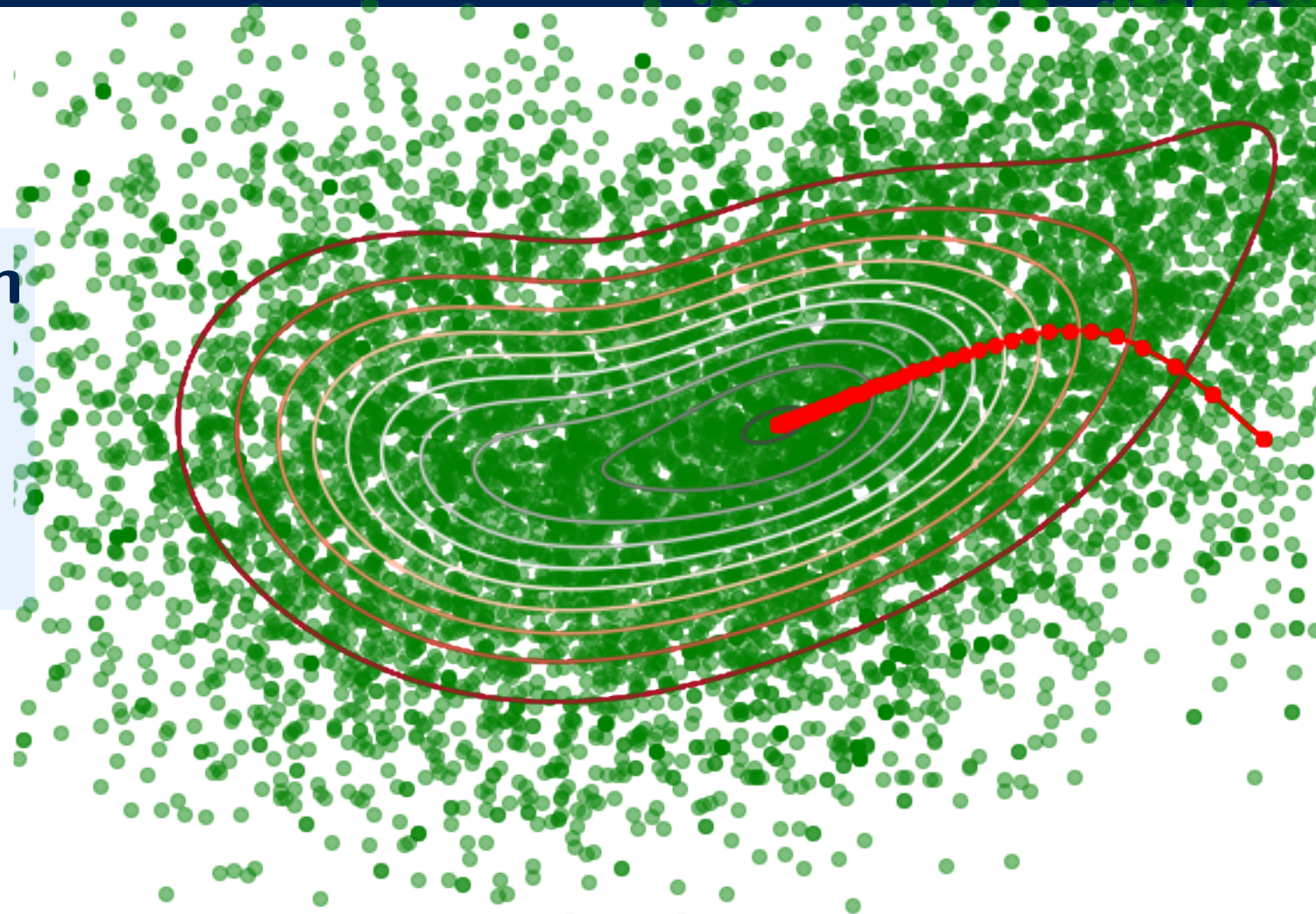
Unadjusted Langevin Algorithm (ULA)

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

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. $\mathcal{N}(0, \text{Id})$



- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.
- Euler-Maruyama discretization of the Langevin eq.
- **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 ∞
- $\|\pi - \pi_\delta\|_{\text{TV}} \leq C\sqrt{\delta}$ (+ W_2 results) [Durmus, Moulines, 2017], [Durmus, De Bortoli, 2020]

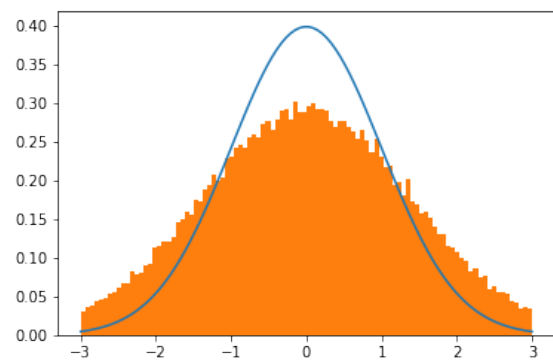
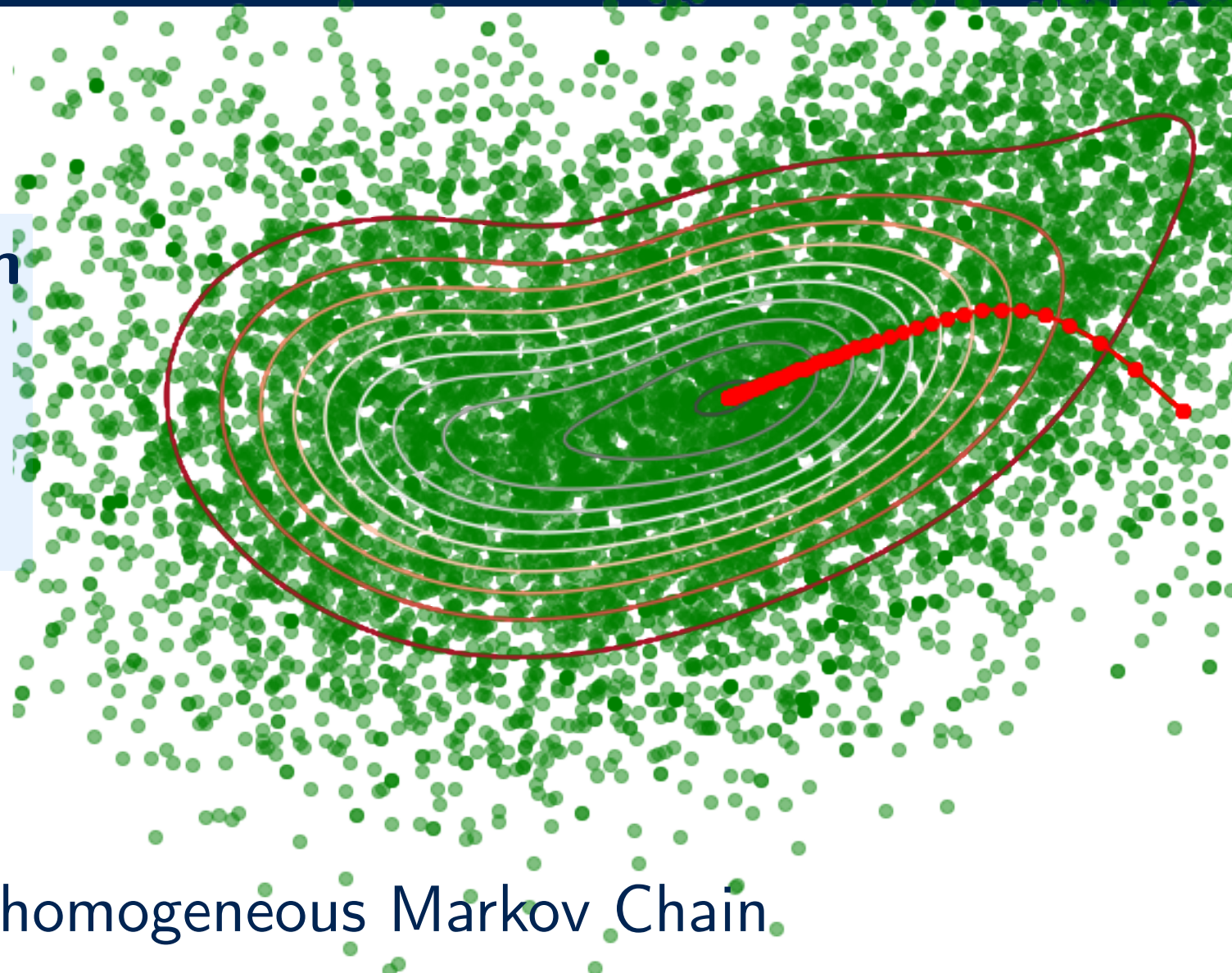
Unadjusted Langevin Algorithm (ULA)

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

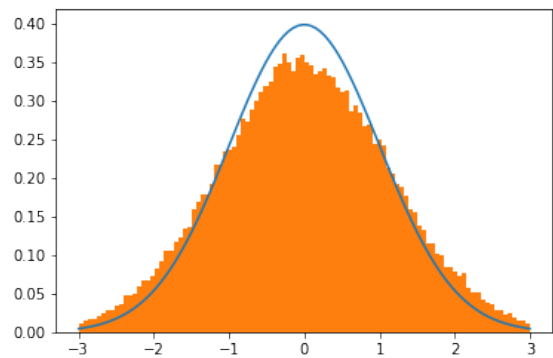
Unadjusted Langevin Algorithm

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

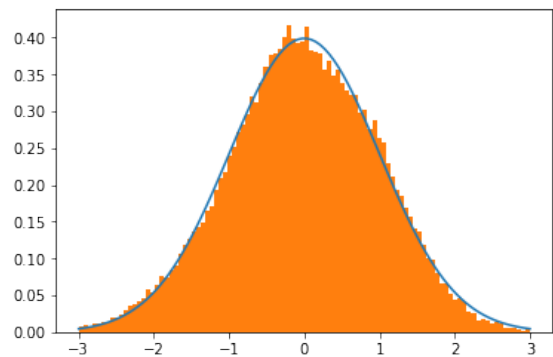
with $(Z_k)_{k \geq 0}$ i.i.d. $\mathcal{N}(0, \text{Id})$



$\delta = 1$



$\delta = 0.5$



$\delta = .01$

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

- $(X_k)_{k \geq 0}$ homogeneous Markov Chain.
- Euler-Maruyama discretization of the Langevin eq.
- **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 ∞
- $\|\pi - \pi_\delta\|_{\text{TV}} \leq C\sqrt{\delta}$ (+ W_2 results) [Durmus, Moulines, 2017], [Durmus, De Bortoli, 2020]

ULA for inverse problems in imaging

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. $\mathcal{N}(0, \text{Id})$

Target density

$$\pi(x) \propto e^{-F(x,y)-U(x)}$$

Hyp: F and U convex, smooth and grad-Lipschitz.

ULA for inverse problems in imaging

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. $\mathcal{N}(0, \text{Id})$

Target density

$$\pi(x) \propto e^{-F(x,y)-U(x)}$$

Hyp: F and U convex, smooth and grad-Lipschitz.

MYULA [Durmus et al. 2018]

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

regularised version of TV
(Moreau-Yosida, or TV-Huber)

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

ULA for inverse problems in imaging

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. $\mathcal{N}(0, \text{Id})$

Target density

$$\pi(x) \propto e^{-F(x,y)-U(x)}$$

Hyp: F and U convex, smooth and grad-Lipschitz.

MYULA [Durmus et al. 2018]

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

regularised version of TV
(Moreau-Yosida, or TV-Huber)

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



50% missing pixels



reconstructed posterior mean



posterior std

A data driven ULA?

ULA

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

A data driven ULA?

ULA

Score function $-\nabla \log p_X$

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

A data driven ULA?

ULA

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

Tweedie's formula [Efron et al. 2011] if $X \sim p_X$ and $Z|X \sim \mathcal{N}(X, \varepsilon I)$, then

$$\nabla \log(p_X * g_\varepsilon)(x) = \frac{D_\varepsilon^\star(x) - x}{\varepsilon}, \text{ where } D_\varepsilon^\star := \mathbb{E}(X|Z = \cdot)$$

A data driven ULA?

ULA

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

Tweedie's formula [Efron et al. 2011] if $X \sim p_X$ and $Z|X \sim \mathcal{N}(X, \varepsilon I)$, then

$$\nabla \log(p_X * g_\varepsilon)(x) = \frac{D_\varepsilon^\star(x) - x}{\varepsilon}, \text{ where } D_\varepsilon^\star := \mathbb{E}(X|Z = .)$$

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

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

A data driven ULA?

ULA

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

Tweedie's formula [Efron et al. 2011] if $X \sim p_X$ and $Z|X \sim \mathcal{N}(X, \varepsilon I)$, then

$$\nabla \log(p_X * g_\varepsilon)(x) = \frac{D_\varepsilon^\star(x) - x}{\varepsilon}, \text{ where } D_\varepsilon^\star := \mathbb{E}(X|Z = .)$$

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

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

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

A data driven ULA?

ULA

$$X_{k+1} = X_k - \delta \nabla F(X_k, y) - \delta \nabla U(X_k) + \sqrt{2\delta} Z_{k+1}, \text{ with } (Z_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \text{Id})$$

Tweedie's formula [Efron et al. 2011] if $X \sim p_X$ and $Z|X \sim \mathcal{N}(X, \varepsilon I)$, then

$$\nabla \log(p_X * g_\varepsilon)(x) = \frac{D_\varepsilon^\star(x) - x}{\varepsilon}, \text{ where } D_\varepsilon^\star := \mathbb{E}(X|Z = .)$$

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

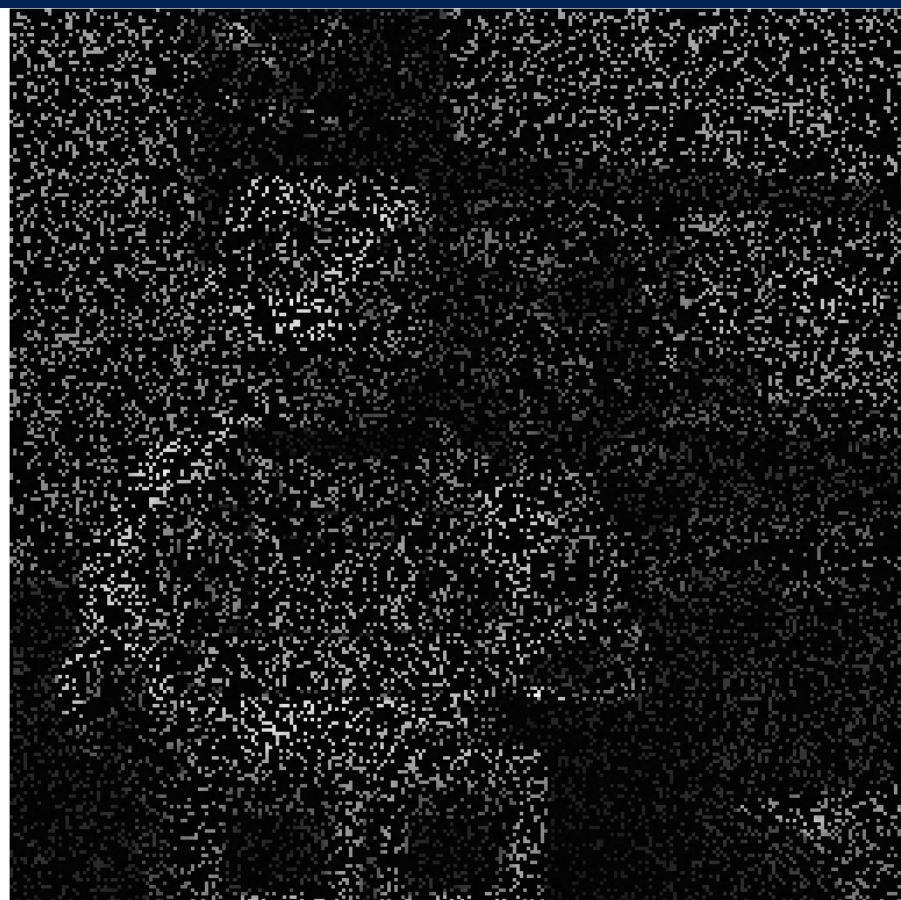
with $(Z_k)_{k \geq 0}$ i.i.d. $\mathcal{N}(0, \text{Id})$

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

- **Versatility/flexibility:** same advantages as proximal PnP
- **Implicit representation:** the true MMSE $D_\varepsilon^\star$ is not known. PnP-ULA rely on denoisers D_ε assumed to be a good approximation of it. **Makes sense for CNN denoisers** (trained to minimize the quadratic risk).



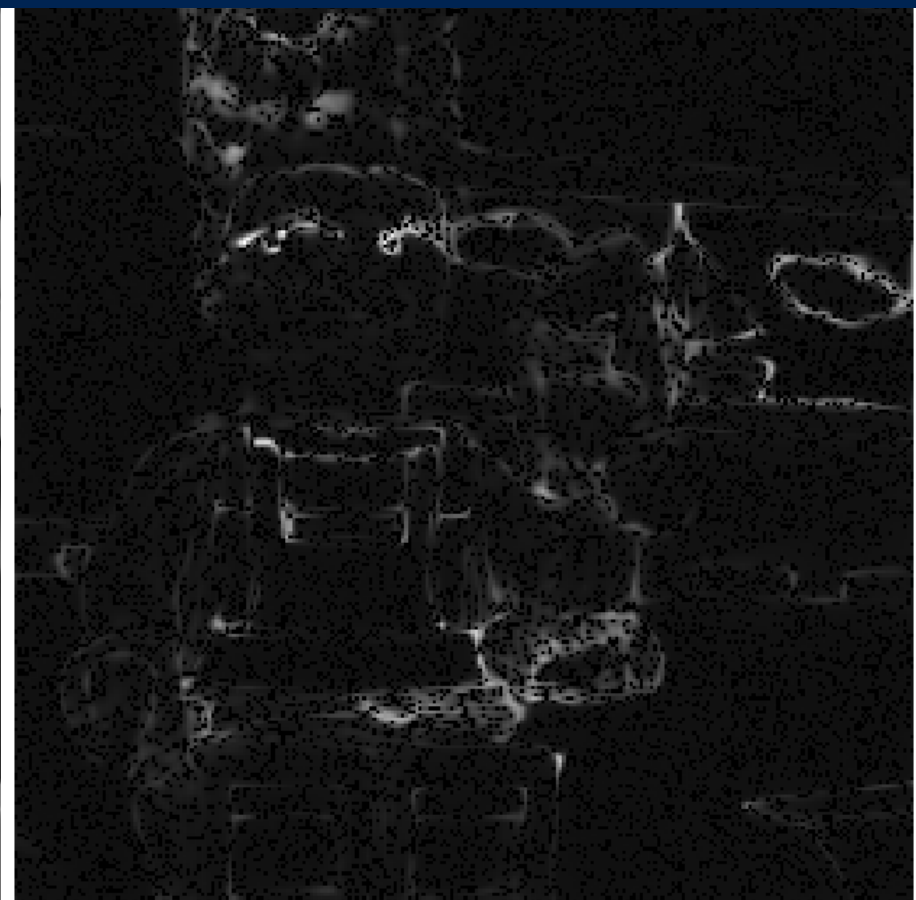
PnP-ULA for inpainting



80% missing pixels



PnP-ULA posterior mean



posterior std

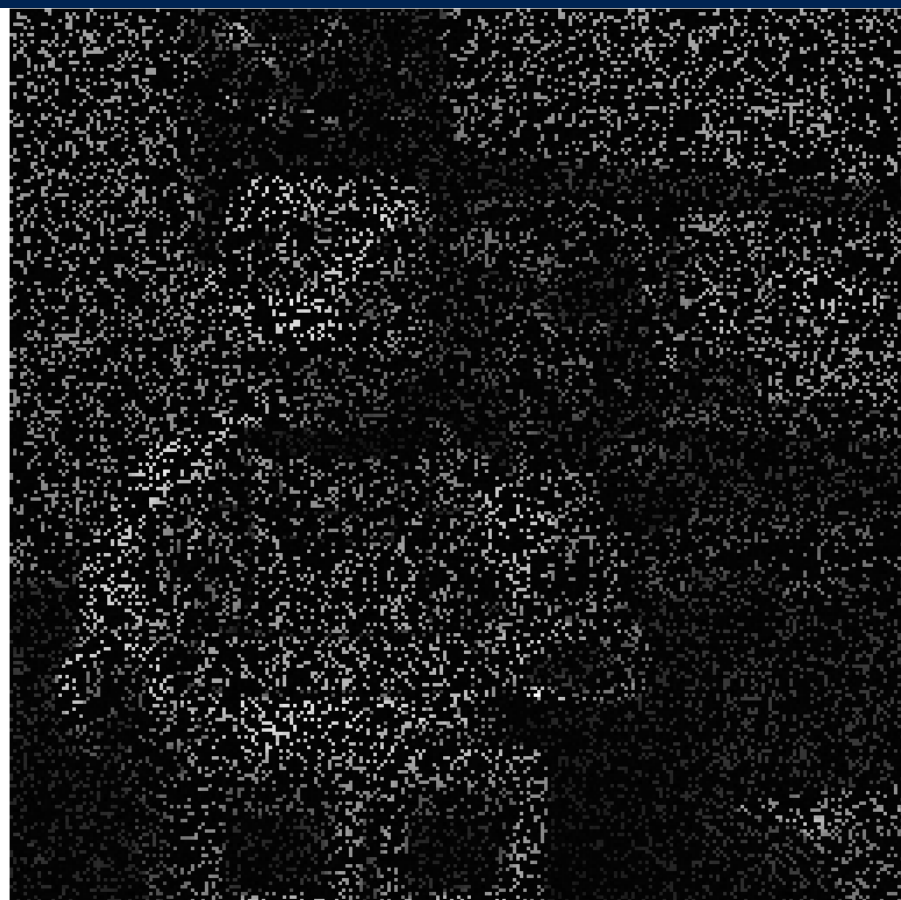


MAP



samples

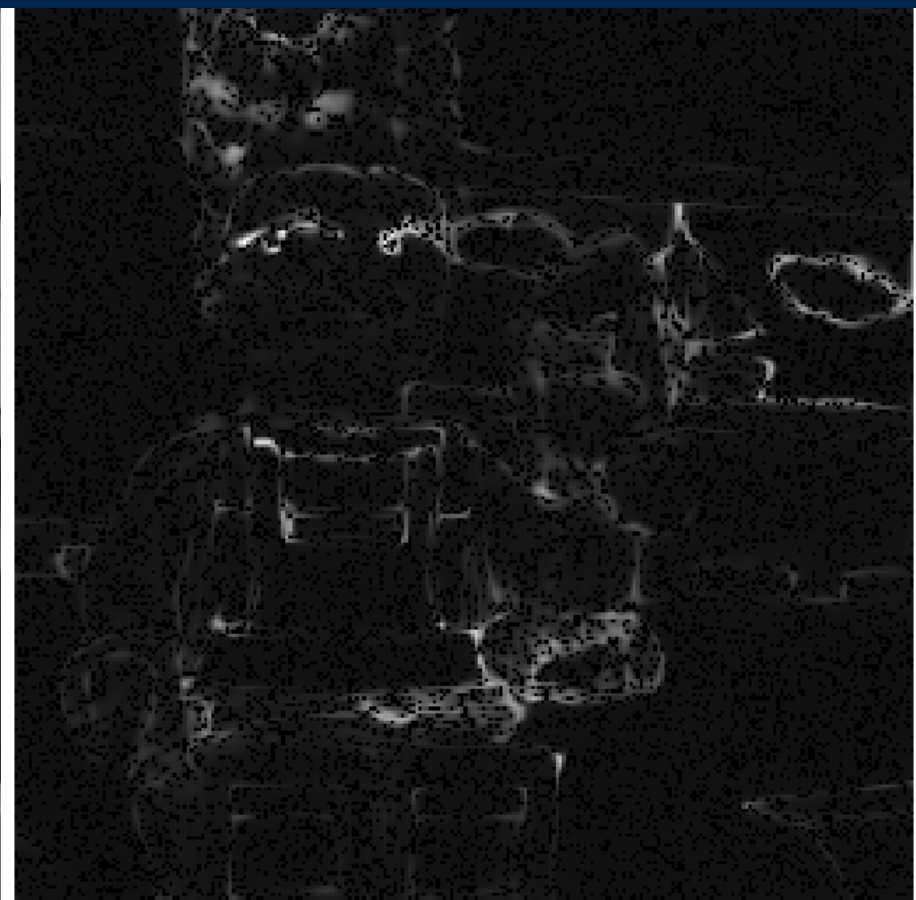
PnP-ULA for inpainting



80% missing pixels



PnP-ULA posterior mean



posterior std



MAP



samples

