“Empirical Bayesian image restoration by Langevin sampling with a denoising diffusion implicit prior”

C. K. Mbakam, J.-F. Giovannelli, M. Pereyra, arXiv:2409.04384

DPP Reading Group

December 4, 2024

Barbara Pascal

Outline

- To get started
  - Inverse problems in a Bayesian framework
  - Langevin sampling
- Denoising Diffusion Probabilistic Models
  - Principles
  - Implicit models
- Plug & Play algorithms
  - Principles
  - Latent-space PnP-Unadjusted Langevin Algorithm

To get started

Observation model

$$\boldsymbol{y} \sim \mathcal{B}\left(\mathbf{A}\boldsymbol{x}^{\star}\right)$$

- $\boldsymbol{y} \in \mathbb{R}^P$: degraded observations;
- $\boldsymbol{x}^{\star} \in \mathbb{R}^N$: unknown quantity of interest;
- $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^P$: known deformation;
- $\mathcal{B}$: random measurement noise.

Observation model

$$\boldsymbol{y} \sim \mathcal{B}\left(\mathbf{A}\boldsymbol{x}^{\star}\right)$$

- $\boldsymbol{y} \in \mathbb{R}^P$: degraded observations;
- $\boldsymbol{x}^{\star} \in \mathbb{R}^N$: unknown quantity of interest;
- $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^P$: known deformation;
- $\mathcal{B}$: random measurement noise.

Goal: Estimate $\boldsymbol{x}^{\star}$

Observation model

$$y \sim \mathcal{B}(\mathbf{A}x^\star)$$

- $y \in \mathbb{R}^P$: degraded observations;
- $x^\star \in \mathbb{R}^N$: unknown quantity of interest;
- $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^P$: known deformation;
- $\mathcal{B}$: random measurement noise.

Goal: Estimate $x^\star$

▶ ill-conditioned, rank deficient $\mathbf{A}$

Inpainting

(Guillemot et al., 2013, *IEEE Sig. Process. Mag.*)

Super-resolution

(Marquina et al., 2008, *J. Sci. Comput.*)

Deblurring

(Pan, 2016, *IEEE Trans. Pattern Anal. Mach. Intell.*)

Inverse problems in a Bayesian framework

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\implies \text{likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Inverse problems in a Bayesian framework

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^{\star} + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\implies \text{likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^{\star}$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\implies$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\implies \text{likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^\star$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\implies$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

$\implies$ prior $p(\boldsymbol{x})$: informative marginal distribution of the random variable $\boldsymbol{x}$

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\Longrightarrow \text{ likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^\star$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\Longrightarrow$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

$\Longrightarrow$ prior $p(\boldsymbol{x})$: informative marginal distribution of the random variable $\boldsymbol{x}$

Bayes theorem: a posteriori distribution of $\boldsymbol{x}|\boldsymbol{y}$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \quad \text{with} \quad p(\boldsymbol{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\Longrightarrow \text{ likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^\star$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\Longrightarrow$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

$\Longrightarrow$ prior $p(\boldsymbol{x})$: informative marginal distribution of the random variable $\boldsymbol{x}$

Bayes theorem: a posteriori distribution of $\boldsymbol{x}|\boldsymbol{y}$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \quad \text{with} \quad p(\boldsymbol{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

Bayesian estimators:

- Maximum A Posteriori (MAP): $\widehat{\boldsymbol{x}} \in \operatorname{Argmax}_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y})$
- Mean A Posteriori (MMSE): $\widehat{\boldsymbol{x}} = \int_{\mathbb{R}^d} \boldsymbol{x} \cdot p(\boldsymbol{x}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}$
- Credibility Region: $\mathbb{P}\left[\boldsymbol{x} \in \mathcal{C}_\alpha\right] = 1 - \alpha$

C. P. Robert, *The Bayesian choice: a decision-theoretic motivation*, 1994

Inverse problems in a Bayesian framework

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\Longrightarrow \text{ likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^\star$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\Longrightarrow$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

$\Longrightarrow$ prior $p(\boldsymbol{x})$: informative marginal distribution of the random variable $\boldsymbol{x}$

Bayes theorem: a posteriori distribution of $\boldsymbol{x}|\boldsymbol{y}$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \quad \text{with} \quad p(\boldsymbol{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

Estimation and uncertainty quantification in a Bayesian framework:

need to sample under the a posteriori distribution $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$

which is **intractable**, in particular in high dimensional problems $d \gg 1$.

Inverse problems in a Bayesian framework

Linear deformation and additive Gaussian noise: $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$

$$\Longrightarrow \text{ likelihood function: } \ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x}) = C \cdot \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2}{2\sigma^2}\right)$$

Estimate $\boldsymbol{x}^\star$ from $\boldsymbol{y}$: usually ill-conditioned or ill-posed $\Longrightarrow$ maximize $\ell_{\boldsymbol{y}}(\boldsymbol{x})$ not enough

$\Longrightarrow$ prior $p(\boldsymbol{x})$: informative marginal distribution of the random variable $\boldsymbol{x}$

Bayes theorem: a posteriori distribution of $\boldsymbol{x}|\boldsymbol{y}$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \quad \text{with} \quad p(\boldsymbol{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

Estimation and uncertainty quantification in a Bayesian framework:

need to sample under the a posteriori distribution $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$

which is **intractable**, in particular in high dimensional problems $d \gg 1$.

Langevin sampling in a nutshell

Target distribution: $\pi(\boldsymbol{x}) = C \cdot \exp(-f(\boldsymbol{x}))$

- $f : \mathbb{R}^d \to \mathbb{R}$ differentiable
- C a (possibly) intractable normalizing constant

Goal: sample under π, i.e., generate realizations of $\boldsymbol{x} \sim \pi$

Langevin sampling in a nutshell

Target distribution: $\pi(\boldsymbol{x}) = C \cdot \exp(-f(\boldsymbol{x}))$

- $f : \mathbb{R}^d \to \mathbb{R}$ differentiable
- C a (possibly) intractable normalizing constant

Goal: sample under π, i.e., generate realizations of $\boldsymbol{x} \sim \pi$

Langevin Stochastic Differential Equation (SDE): $\mathbf{B}_t$ Wiener process in $\mathbb{R}^d$

$$\mathbf{X}_0 \in \mathbb{R}^d, \quad \mathrm{d}\mathbf{X}_t = \nabla \log \pi(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

Target distribution: $\pi(\boldsymbol{x}) = C \cdot \exp(-f(\boldsymbol{x}))$

- $f : \mathbb{R}^d \to \mathbb{R}$ differentiable
- C a (possibly) intractable normalizing constant

Goal: sample under π, i.e., generate realizations of $\boldsymbol{x} \sim \pi$

Langevin Stochastic Differential Equation (SDE): $\mathbf{B}_t$ Wiener process in $\mathbb{R}^d$

$$\mathbf{X}_0 \in \mathbb{R}^d, \quad \mathrm{d}\mathbf{X}_t = \nabla \log \pi(\mathbf{X}_t)\, \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

If π is proper and smooth, with $\nabla \log \pi$ Lipschitz-continuous, i.e., $\exists L > 0$ such that

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d, \quad \|\nabla \log \pi(\boldsymbol{x}) - \nabla \log \pi(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$$

$\Longrightarrow$ unique solution $(\mathbf{X}_t)_{t\geq 0}$ with π as unique stationary density: for $T \gg 1$, $\mathbf{X}_T \sim \pi$.

J. Kent, *Adv Appl Probab*, 1978; G. O. Roberts & R. L. Tweedie, *Bernoulli*, 1996

Target distribution: $\pi(\boldsymbol{x}) = C \cdot \exp(-f(\boldsymbol{x}))$

- $f : \mathbb{R}^d \to \mathbb{R}$ differentiable
- C a (possibly) intractable normalizing constant

Goal: sample under π, i.e., generate realizations of $\boldsymbol{x} \sim \pi$

Langevin Stochastic Differential Equation (SDE): $\mathbf{B}_t$ Wiener process in $\mathbb{R}^d$

$$\mathbf{X}_0 \in \mathbb{R}^d, \quad \mathrm{d}\mathbf{X}_t = \nabla \log \pi(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

Example: sampling a Gaussian $\pi(\boldsymbol{x}) = C \cdot \exp(-\|\boldsymbol{x}\|_2^2/2)$

(Ornstein-Ulhenbeck process) $\quad \mathbf{X}_0 \in \mathbb{R}^d, \quad \mathrm{d}\mathbf{X}_t = -\mathbf{X}_t\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$

$\Longrightarrow$ solution: $\mathbf{X}_t \sim \mathrm{e}^{-t}\mathbf{X}_0 + \sqrt{1-\mathrm{e}^{-2t}}\,\mathbf{Z}$, where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

If $\mathbf{X}_0 \sim \pi_0$, then $\mathbf{X}_t \sim \pi_t$ with $\pi_t = \pi_0(\cdot/\mathrm{e}^t) * \mathcal{N}(\mathbf{0}, 1-\mathrm{e}^{-2t})$:

Gaussian smoothing with increasing bandwidth.

Langevin sampling in a nutshell

Target distribution: $\pi(\boldsymbol{x}) = C \cdot \exp(-f(\boldsymbol{x}))$

- $f : \mathbb{R}^d \to \mathbb{R}$ differentiable
- C a (possibly) intractable normalizing constant

Goal: sample under π, i.e., generate realizations of $\boldsymbol{x} \sim \pi$

Langevin Stochastic Differential Equation (SDE): $\mathbf{B}_t$ Wiener process in $\mathbb{R}^d$

$$\mathbf{X}_0 \in \mathbb{R}^d, \quad \mathrm{d}\mathbf{X}_t = \nabla \log \pi(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

In general, no closed-form solution $\Longrightarrow$ Euler-Maruyama discretization

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \tau \nabla \log \pi(\mathbf{X}_k) + \sqrt{2\tau}\mathbf{Z}_k, \quad \mathbf{Z}_k \text{ i.i.d. standard Gaussian vectors}$$

Unadjusted Langevin Algorithm (ULA) converges to π up to **discretization bias**.

MALA removes the bias at the expense of a Metropolis-Hastings correction step.

A. S. Dalalyan, *J. R. Stat. Soc.* , 2017; G. O. Roberts & R. L. Tweedie, *Bernoulli*, 1996

Inverse problem: estimate $\boldsymbol{x}$ from observations $\boldsymbol{y} = \mathcal{D}\left(\mathbf{A}(\boldsymbol{x})\right)$

- likelihood function: $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})$
- prior: $p(\boldsymbol{x})$

$\Longrightarrow$ a posteriori distribution $p(\boldsymbol{x}|\boldsymbol{y}) = \dfrac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \quad p(\boldsymbol{y}) = \displaystyle\int_{\mathbb{R}^d} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$

Unadjusted Langevin Algorithm: $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \tau\nabla \log p(\boldsymbol{y}|\mathbf{X}_k) + \tau\nabla \log p(\mathbf{X}_k) + \sqrt{2\tau}\mathbf{Z}_k$$

$\mathbf{Z}_k$ i.i.d. standard Gaussian vectors.

Bayesian estimators:

- Maximum A Posteriori (MAP): $\widehat{\boldsymbol{x}} \simeq \mathrm{Argmax}\ \{p(\boldsymbol{y}|\mathbf{X}_k)p(\mathbf{X}_k),\ k \geq K_{\text{burnin}}\}$
- Mean A Posteriori (MMSE): $\widehat{\boldsymbol{x}} \simeq \mathrm{Mean}\ \{\mathbf{X}_k,\ k \geq K_{\text{burnin}}\}$
- Credibility Region: $\widehat{\mathcal{C}}_\alpha \sim \mathrm{Quantiles}_\alpha\ \{\mathbf{X}_k,\ k \geq K_{\text{burnin}}\}$

Bayesian framework: aims at sampling under $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ from two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓

To get started - Main bottleneck and a research path

Bayesian framework: aims at sampling under $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ from two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a"realistic" $\boldsymbol{x}^{\star}$ ✗

To get started - Main bottleneck and a research path

Bayesian framework: aims at sampling under $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ from two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a"realistic" $\boldsymbol{x}^\star$ ✗

Prior design: key to obtain accurate estimates

- sparsity in transformed domain or piecewise regularity $p(\boldsymbol{x}) \propto \mathrm{e}^{-\mu\|\mathbf{L}\boldsymbol{x}\|_1}$
- Markov random fields
- learned patch-based Gaussian or Gaussian mixture models

$\Longrightarrow$ not reflecting the diversity and complexity of true images

To get started - Main bottleneck and a research path

Bayesian framework: aims at sampling under $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ from two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a"realistic" $\boldsymbol{x}^\star$ ✗

Prior design: key to obtain accurate estimates

- sparsity in transformed domain or piecewise regularity $p(\boldsymbol{x}) \propto \mathrm{e}^{-\mu\|\mathbf{L}\boldsymbol{x}\|_1}$
- Markov random fields
- learned patch-based Gaussian or Gaussian mixture models

$\Longrightarrow$ not reflecting the diversity and complexity of true images

Data-driven prior: learn, possibly implicitly, either

- the posterior $\pi(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ from a set of pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$
- or, the prior $\pi_0(\boldsymbol{x}) = p(\boldsymbol{x})$ from a collection of training samples $\{\boldsymbol{x}_i\}_{i=1}^N$

Denoising Diffusion Probabilistic Models

Diffusion models

Goal: learn to sample from a distribution π_0 with **no explicit expression** from

a collection of **samples** $\{x_i\}_{i=1}^{N}$, $x_i \sim \pi_0$

Goal: learn to sample from a distribution π_0 with **no explicit expression** from

a collection of **samples** $\{x_i\}_{i=1}^N$, $x_i \sim \pi_0$

Ornstein-Uhlenbeck process in $\mathbb{R}^d$: stochastic differential equation

$$\mathrm{d}\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t$$

- $\mathbf{W}$: Brownian motion in dimension d
- β: positive weighting function

from $\mathbf{X}_0 \sim \pi_0$ to $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Goal: learn to sample from a distribution π_0 with **no explicit expression** from a collection of **samples** $\{x_i\}_{i=1}^N$, $x_i \sim \pi_0$

Ornstein-Uhlenbeck process in $\mathbb{R}^d$: stochastic differential equation

$$\mathrm{d}\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t$$

- $\mathbf{W}$: Brownian motion in dimension d
- β: positive weighting function

from $\mathbf{X}_0 \sim \pi_0$ to $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Backward process in $\mathbb{R}^d$: reversed stochastic differential equation $\mathbf{X}_t \sim \pi_t$

$$\mathrm{d}\mathbf{X}_t = \left[-\frac{1}{2}\beta(t)\mathbf{X}_t - \beta(t)\nabla \log \pi_t(\mathbf{X}_t)\right] \mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t$$

- $\overline{\mathbf{W}}$: Brownian motion in dimension d
- t flowing backward from ∞ to $t = 0$

from $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to $\mathbf{X}_0 \sim \pi_0$

Goal: learn to sample from a distribution π_0 with **no explicit expression** from a collection of **samples** $\{x_i\}_{i=1}^N$, $x_i \sim \pi_0$

Ornstein-Uhlenbeck process in $\mathbb{R}^d$: stochastic differential equation

$$\mathrm{d}\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t$$

- $\mathbf{W}$: Brownian motion in dimension d
- β: positive weighting function

from $\mathbf{X}_0 \sim \pi_0$ to $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Backward process in $\mathbb{R}^d$: reversed stochastic differential equation $\mathbf{X}_t \sim \pi_t$

$$\mathrm{d}\mathbf{X}_t = \left[-\frac{1}{2}\beta(t)\mathbf{X}_t - \beta(t)\nabla \log \pi_t(\mathbf{X}_t)\right]\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t$$

- $\overline{\mathbf{W}}$: Brownian motion in dimension d
- t flowing backward from ∞ to $t = 0$

from $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to $\mathbf{X}_0 \sim \pi_0$

Diffusion models

Goal: learn to sample from a distribution π_0 with **no explicit expression** from

a collection of **samples** $\{x_i\}_{i=1}^N$, $x_i \sim \pi_0$

How to use a diffusion model:

- learn score functions $\boldsymbol{x} \mapsto \nabla \log \pi_t(\boldsymbol{x})$ via score matching from samples $\{\boldsymbol{x}_i\}_{i=1}^N$,
- draw $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is easy,
- solve $\mathrm{d}\mathbf{X}_t = \left[-\frac{1}{2}\beta(t)\mathbf{X}_t - \beta(t)\nabla \log \pi_t(\mathbf{X}_t)\right] \mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t$ from ∞ to 0.

Goal: learn to sample from a distribution π_0 with **no explicit expression** from

a collection of **samples** $\{x_i\}_{i=1}^N$, $x_i \sim \pi_0$

How to use a diffusion model:

- learn score functions $\boldsymbol{x} \mapsto \nabla \log \pi_t(\boldsymbol{x})$ via score matching from samples $\{\boldsymbol{x}_i\}_{i=1}^N$,
- draw $\mathbf{X}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is easy,
- solve $\mathrm{d}\mathbf{X}_t = \left[-\frac{1}{2}\beta(t)\mathbf{X}_t - \beta(t)\nabla \log \pi_t(\mathbf{X}_t)\right] \mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t$ from ∞ to 0.

In practice:

- score functions $\boldsymbol{x} \mapsto \nabla \log \pi_t(\boldsymbol{x})$ approximated by a neural network $\mathbf{s}_\vartheta(\boldsymbol{x}, t)$,
- sample $\mathbf{X}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with T "large",
- approximate the solution of the backward SDE by a discrete-time scheme:

  Denoising Diffusion Probabilistic Models; Denoising Diffusion Implicit Models.

J. Ho et al., *Adv Neural Inf Process Syst*, 2020; J. Song et al., *ICLR*, 2021

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \dots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \dots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\,\mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t}(1-\alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

In practice: $\boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t) = \sigma_t^2 \mathbf{I}$ fixed, e.g., to $\sigma_t^2 = \alpha_t$, not learned.

J. Ho et al., *Adv Neural Inf Process Syst*, 2020; J. Song et al., *ICLR*, 2021

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\,\mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t}(1-\alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

In practice: $\boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t) = \sigma_t^2 \mathbf{I}$ fixed, e.g., to $\sigma_t^2 = \alpha_t$, not learned. Let $\mathbf{X}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{X}_{t-1} = \boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t) + \sigma_t^2 \mathbf{Z}_t, \quad \mathbf{Z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

where $\boldsymbol{\mu}_\vartheta$ can be learned using **score matching** techniques following discretization

$$\mathbf{X}_{t-1} = \mathbf{X}_t + \frac{\tau \beta_t}{2} \mathbf{X}_t + \tau \beta_t \nabla \log \pi_t(\mathbf{X}_t) + \sqrt{\tau \beta_t} \mathbf{Z}_t, \quad \mathbf{Z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

J. Ho et al., *Adv Neural Inf Process Syst*, 2020; J. Song et al., *ICLR*, 2021

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\, \mathbf{X}_0 + \sqrt{1 - \overline{\alpha}_t}\, \mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t} (1 - \alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Denoising Diffusion Implicit Model (DDIM): discrete time $t = 0, 1, \ldots, T$

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\,\mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t}(1-\alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Denoising Diffusion Implicit Model (DDIM): discrete time $t = 0, 1, \ldots, T$

- variance sequences $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$ and $(\sigma_t)_{t=2}^{T}$, $\sigma_t \in [0, \sqrt{1-\alpha_{t-1}}[$

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\,\mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t}(1-\alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Denoising Diffusion Implicit Model (DDIM): discrete time $t = 0, 1, \ldots, T$

- variance sequences $(\alpha_t)_{t=1}^{T}$, $0 \leq \alpha_t < 1$ and $(\sigma_t)_{t=2}^{T}$, $\sigma_t \in [0, \sqrt{1-\alpha_{t-1}}[$
- non-Markovian forward process satisfying

$$\mathbf{X}_{t-1} | \mathbf{X}_t, \mathbf{X}_0 \sim \mathcal{N}\left(\sqrt{\alpha_{t-1}} \cdot \mathbf{X}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{X}_t - \sqrt{\alpha_t}\mathbf{X}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \mathbf{I}\right)$$

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\, \mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\, \mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t} (1-\alpha_s)$

Denoising Diffusion Probabilistic Model (DDPM): discrete time $t = 0, 1, \ldots, T$

- variance sequence $(\alpha_t)_{t=1}^T$, $0 \leq \alpha_t < 1$
- forward process $\mathbf{X}_t | \mathbf{X}_{t-1} \sim \mathcal{N}\left(\sqrt{1-\alpha_t}\mathbf{X}_{t-1}, \alpha_t \mathbf{I}\right)$
- backward process $\mathbf{X}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_\vartheta(\mathbf{X}_t, t), \boldsymbol{\Sigma}_\vartheta(\mathbf{X}_t, t)\right)$ from learned $\boldsymbol{\mu}_\vartheta, \boldsymbol{\Sigma}_\vartheta$

Denoising Diffusion Implicit Model (DDIM): discrete time $t = 0, 1, \ldots, T$

- variance sequences $(\alpha_t)_{t=1}^T$, $0 \leq \alpha_t < 1$ and $(\sigma_t)_{t=2}^T$, $\sigma_t \in [0, \sqrt{1-\alpha_{t-1}}[$
- non-Markovian forward process satisfying

$$\mathbf{X}_{t-1} | \mathbf{X}_t, \mathbf{X}_0 \sim \mathcal{N}\left(\sqrt{\alpha_{t-1}} \cdot \mathbf{X}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{X}_t - \sqrt{\alpha_t}\mathbf{X}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \mathbf{I}\right)$$

- $\mathrm{f}_\vartheta(\boldsymbol{x}_t)$ learnable predictor of $\mathbf{X}_0$ from $\mathbf{X}_t$: backward $\mathbf{X}_{t-1} | \mathrm{f}_\vartheta(\mathbf{X}_t), \mathbf{X}_0$

Gaussian marginals: $\mathbf{X}_t = \sqrt{\overline{\alpha}_t}\, \mathbf{X}_0 + \sqrt{1-\overline{\alpha}_t}\, \mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \overline{\alpha}_t = \prod_{s=1}^{t} (1 - \alpha_s)$

Plug & Play principles applied to sampling algorithms

Plug-and-Play: motivations

Bayesian framework: two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓

Bayesian framework: two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a"realistic" $\boldsymbol{x}^\star$ ✗

Plug-and-Play: motivations

Bayesian framework: two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a"realistic" $\boldsymbol{x}^\star$ ✗

Prior design: key to obtain accurate estimates

- sparsity in transformed domain or piecewise regularity $p(\boldsymbol{x}) \propto \mathrm{e}^{-\mu\|\mathbf{L}\boldsymbol{x}\|_1}$
- Markov random fields
- learned patch-based Gaussian or Gaussian mixture models

$\Longrightarrow$ not reflecting the diversity and complexity of true images

Bayesian framework: two ingredients

- likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ observation modeling: deformation and noise ✓
- prior $p(\boldsymbol{x})$: expected characteristics of a“realistic” $\boldsymbol{x}^\star$ ✗

Prior design: key to obtain accurate estimates

- sparsity in transformed domain or piecewise regularity $p(\boldsymbol{x}) \propto \mathrm{e}^{-\mu\|\mathbf{L}\boldsymbol{x}\|_1}$
- Markov random fields
- learned patch-based Gaussian or Gaussian mixture models

$\Longrightarrow$ not reflecting the diversity and complexity of true images

Plug & Play: learn $\pi_0(\boldsymbol{x}) = p(\boldsymbol{x})$ from a collection of training samples $\{\boldsymbol{x}_i\}$

then interpret $\lambda\nabla_{\boldsymbol{x}} \log p_\lambda(\boldsymbol{x}_k)$ as a **denoising** correction $\mathrm{D}_\lambda(\boldsymbol{x}) - \boldsymbol{x}$

$\mathrm{D}_\lambda(\boldsymbol{x})$: denoising operator trained to remove Gaussian noise of variance λ.

S. V. Venkatakrishnan et al., *IEEE Glob. Conf. Signal Inf. Process.*, 2013

Unadjusted Langevin Algorithm (ULA): to sample $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log p(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Unadjusted Langevin Algorithm (ULA): to sample $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log p(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Plug & Play ULA (PnP-ULA): explicit likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log \underset{?}{\pi_0}(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

$\Longrightarrow$ π_0 is an **implicit** probability density: might even not be proper!

Unadjusted Langevin Algorithm (ULA): to sample $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log p(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Plug & Play ULA (PnP-ULA): explicit likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log \underset{?}{\pi_0}(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

$\Longrightarrow$ π_0 is an **implicit** probability density: might even not be proper!

Instead, marginal regularized density $p_\lambda(\mathbf{X}_\lambda)$ associated to $\mathbf{X}_\lambda \sim \mathcal{N}(\mathbf{X}, \lambda \mathbf{I})$, $\mathbf{X} \sim \pi_0$

$$\textbf{Tweedie's formula:}\ \nabla_{\boldsymbol{x}} \log p_\lambda(\boldsymbol{x}) \simeq \frac{\mathrm{D}^\star_\lambda(\boldsymbol{x}) - \boldsymbol{x}}{\lambda},$$

$\mathrm{D}^\star_\lambda$ the Minimal Mean Squared Error (MMSE) denoiser to find $\mathbf{X}$ from $\mathbf{X}_\lambda$:

$$\mathrm{D}^\star_\lambda \in \underset{\mathrm{D}_\lambda}{\mathrm{Argmin}}\ \mathbb{E}_{\mathbf{X}\sim\pi, \mathbf{X}_\lambda \sim \mathcal{N}(\mathbf{X}, \lambda \mathbf{I})} \|\mathrm{D}_\lambda(\mathbf{X}_\lambda) - \mathbf{X}\|_2^2$$

Unadjusted Langevin Algorithm (ULA): to sample $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log p(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Plug & Play ULA (PnP-ULA): explicit likelihood $\ell_{\boldsymbol{y}}(\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \nabla_{\boldsymbol{x}} \log \underset{?}{\pi_0}(\mathbf{X}_k) + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

$\Longrightarrow$ π_0 is an **implicit** probability density: might even not be proper!

$\widehat{\mathrm{D}}_\lambda$ an approximate MMSE estimator, e.g., trained neural network, plugged in ULA:

$$\Longrightarrow \mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\widehat{\mathrm{D}}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

ensure convergence by enforcing Lipschitzianity during training

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d, \quad \|\nabla \log \widehat{\mathrm{D}}_\lambda(\boldsymbol{x}) - \nabla \log \widehat{\mathrm{D}}_\lambda(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$$

Laumont et al., *SIAM J. Imaging Sci.*, 2022; E. K. Ryu, *Proc. Int. Conf. Mach. Learn.*, 2019

Plug-and-Play Unadjusted Langevin Algorithm: (PnP-ULA)

- solid convergence guarantees ✓
- high-fidelity to the data through explicit likelihood ✓
- once stationarity reached, efficient to generate new samples ✓
- fine details lost due to Lipschitz regularization during training ✗

Plug-and-Play Unadjusted Langevin Algorithm: (PnP-ULA)

- solid convergence guarantees ✓
- high-fidelity to the data through explicit likelihood ✓
- once stationarity reached, efficient to generate new samples ✓
- fine details lost due to Lipschitz regularization during training ✗

Idea: Use as denoiser the last layers of a **diffusion model**

$\boldsymbol{\Psi}_{\beta}(\boldsymbol{x})$: Markov kernel associated to a single reversed diffusion step at level β.

Removing additive Gaussian noise of variance $\lambda = \beta_{t^\star}$ from $\mathbf{X} \mid \mathbf{X}_0 \sim \mathcal{N}(\mathbf{X}_0, \lambda \boldsymbol{I})$:

$$\widetilde{\mathrm{D}}_{\lambda}(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x}).$$

Plug-and-Play Unadjusted Langevin Algorithm: (PnP-ULA)

- solid convergence guarantees ✓
- high-fidelity to the data through explicit likelihood ✓
- once stationarity reached, efficient to generate new samples ✓
- fine details lost due to Lipschitz regularization during training ✗

Idea: Use as denoiser the last layers of a **diffusion model**

$\boldsymbol{\Psi}_\beta(\boldsymbol{x})$: Markov kernel associated to a single reversed diffusion step at level β.

Removing additive Gaussian noise of variance $\lambda = \beta_{t^\star}$ from $\mathbf{X} \mid \mathbf{X}_0 \sim \mathcal{N}(\mathbf{X}_0, \lambda \boldsymbol{I})$:

$$\widetilde{\mathrm{D}}_\lambda(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x}).$$

Problem: $\widetilde{\mathrm{D}}_\lambda$ possibly **not Lipschitzian** $\Longrightarrow$ reconstruction artifacts

cause of **divergence** of the Plug-and-Play Unadjusted Algorithm.

Plug-and-Play ULA using Diffusion Models: $\widetilde{\mathrm{D}}_\lambda(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\widetilde{\mathrm{D}}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Equivariant Plug-and-Play Unadjusted Langevin Algorithm

Plug-and-Play ULA using Diffusion Models: $\widetilde{\mathrm{D}}_\lambda(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\widetilde{\mathrm{D}}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Equivariant Plug-and-Play approach: assume equivariance under the action of $\mathcal{G}$

$$\forall g \in \mathcal{G}, \quad \mathbf{X}_0 \sim \pi_0 \Leftrightarrow g \cdot \mathbf{X}_0 \sim \pi_0$$

e.g., group of rotations, small translations, reflections.

In words, "when translated a realistic image is still realistic".

Equivariant Plug-and-Play Unadjusted Langevin Algorithm

Plug-and-Play ULA using Diffusion Models: $\widetilde{\mathrm{D}}_\lambda(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\widetilde{\mathrm{D}}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Equivariant Plug-and-Play approach: assume equivariance under the action of $\mathcal{G}$

$$\forall g \in \mathcal{G}, \quad \mathbf{X}_0 \sim \pi_0 \Leftrightarrow g \cdot \mathbf{X}_0 \sim \pi_0$$

e.g., group of rotations, small translations, reflections.

In words, "when translated a realistic image is still realistic".

Original

Translated with periodic conditions

Equivariant Plug-and-Play Unadjusted Langevin Algorithm

Plug-and-Play ULA using Diffusion Models: $\widetilde{\mathrm{D}}_\lambda(\boldsymbol{x}) = \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(\boldsymbol{x})$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_x \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\widetilde{\mathrm{D}}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Equivariant Plug-and-Play approach: assume equivariance under the action of $\mathcal{G}$

$$\forall g \in \mathcal{G}, \quad \mathbf{X}_0 \sim \pi_0 \Leftrightarrow g \cdot \mathbf{X}_0 \sim \pi_0$$

e.g., group of rotations, small translations, reflections.

In words, "when translated a realistic image is still realistic".

$\mathcal{G}$-equivariant denoiser $\mathrm{D}_\lambda(\boldsymbol{x}) = g^{-1} \cdot \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(g \cdot \boldsymbol{x}), \quad g \sim \mathcal{U}_{\mathcal{G}}$

$\mathcal{U}_{\mathcal{G}}$ uniform distribution on $\mathcal{G} \Longrightarrow$ at each step of PnP-ULA:

random translation – denoising – inverse translation.

M. Terris et al., *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

Latent space Plug-and-Play Unadjusted Langevin Algorithm

Equivariant Plug-and-Play ULA: $\mathsf{D}_\lambda(\boldsymbol{x}) = g^{-1} \cdot \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(g \cdot \boldsymbol{x}), \;\; g \sim \mathcal{U}_{\mathcal{G}}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\mathsf{D}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Latent space Plug-and-Play Unadjusted Langevin Algorithm

Equivariant Plug-and-Play ULA: $\mathrm{D}_\lambda(\boldsymbol{x}) = g^{-1} \cdot \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(g \cdot \boldsymbol{x}), \; g \sim \mathcal{U}_\mathcal{G}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\mathrm{D}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Latent space strategy: $\rho > 0$ and $\mathbf{U}$ and auxiliary random variable so that

$$\mathbf{X} \mid \mathbf{U} \sim \mathcal{N}(\mathbf{U}, \rho \mathbf{I}) \quad \text{and} \quad \nabla_{\boldsymbol{u}} \log p(\boldsymbol{u}) \simeq \frac{\mathrm{D}_\lambda(\boldsymbol{u}) - \boldsymbol{u}}{\lambda}$$

where ρ controls $\|\mathbf{X} - \mathbf{U}\|_2^2 \implies$ accounts for the fact that D_λ is imperfect.

to obtain both high reconstruction accuracy & fast convergence

Latent space Plug-and-Play Unadjusted Langevin Algorithm

Equivariant Plug-and-Play ULA: $\mathrm{D}_\lambda(\boldsymbol{x}) = g^{-1} \cdot \boldsymbol{\Psi}_{\beta_0} \circ \ldots \circ \boldsymbol{\Psi}_{\beta_{t^\star}}(g \cdot \boldsymbol{x}), \;\; g \sim \mathcal{U}_{\mathcal{G}}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma \nabla_{\boldsymbol{x}} \log \ell_{\boldsymbol{y}}(\mathbf{X}_k) + \gamma \frac{\mathrm{D}_\lambda(\mathbf{X}_k) - \mathbf{X}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

Latent space strategy: $\rho > 0$ and $\mathbf{U}$ and auxiliary random variable so that

$$\mathbf{X} \mid \mathbf{U} \sim \mathcal{N}(\mathbf{U}, \rho \mathbf{I}) \quad \text{and} \quad \nabla_{\boldsymbol{u}} \log p(\boldsymbol{u}) \simeq \frac{\mathrm{D}_\lambda(\boldsymbol{u}) - \boldsymbol{u}}{\lambda}$$

where ρ controls $\|\mathbf{X} - \mathbf{U}\|_2^2 \implies$ accounts for the fact that D_λ is imperfect.

to obtain both high reconstruction accuracy & fast convergence

Latent space equivariant Plug-and-Play ULA: targets $\mathbf{U} \mid \mathbf{Y} = \boldsymbol{y}$

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \gamma \nabla_{\boldsymbol{u}} \log p(\boldsymbol{y} | \mathbf{U}_k, \rho) + \gamma \frac{\mathrm{D}_\lambda(\mathbf{U}_k) - \mathbf{U}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

where $\boldsymbol{u} \mapsto p(\boldsymbol{y} | \boldsymbol{u}, \rho)$ is strongly log-concave thanks to $\rho > 0$.

Mean A Posteriori estimators: $\mathrm{Mean}\left\{\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X}),\, k \geq K_{\mathsf{burnin}}\right\}$

$\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X})$ tractable analytically for most φ due to Gaussianity

Mean A Posteriori estimators: $\mathrm{Mean}\left\{\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X}),\, k \geq K_{\text{burnin}}\right\}$

$\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X})$ tractable analytically for most φ due to Gaussianity

Performance depends on ρ: trade-off between accuracy and convergence speed.

Mean A Posteriori estimators: $\mathrm{Mean}\left\{\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X}),\, k \geq K_{\text{burnin}}\right\}$

$\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X})$ tractable analytically for most φ due to Gaussianity

Performance depends on ρ: trade-off between accuracy and convergence speed.

Maximum Marginal Likelihood Estimation of hyperparameters:

$$\widehat{\rho}(\boldsymbol{y}) \in \operatorname*{Argmax}_{\rho>0} p(\boldsymbol{y}|\rho), \quad \text{where} \quad p(\boldsymbol{y}|\rho) = \mathbb{E}_{\mathbf{X},\mathbf{U}|\boldsymbol{y},\rho}\left[\ell_{\boldsymbol{y}}(\mathbf{X})\right]$$

$\Longrightarrow$ **combine** optimization of the marginal likelihood and generation of samples

Mean A Posteriori estimators: $\mathrm{Mean}\left\{\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X}),\, k \geq K_{\mathrm{burnin}}\right\}$

$\mathbb{E}_{\mathbf{X}|\boldsymbol{y},\mathbf{U}_k,\rho}\,\varphi(\mathbf{X})$ tractable analytically for most φ due to Gaussianity

Performance depends on ρ: trade-off between accuracy and convergence speed.

Maximum Marginal Likelihood Estimation of hyperparameters:

$$\widehat{\rho}(\boldsymbol{y}) \in \underset{\rho>0}{\mathrm{Argmax}}\, p(\boldsymbol{y}|\rho), \quad \text{where} \quad p(\boldsymbol{y}|\rho) = \mathbb{E}_{\mathbf{X},\mathbf{U}|\boldsymbol{y},\rho}\left[\ell_{\boldsymbol{y}}(\mathbf{X})\right]$$

$\Longrightarrow$ **combine** optimization of the marginal likelihood and generation of samples

Stochastic Approximation Proximal Gradient: (SAPG) $\rho_0 > 0$ and $\mathbf{U}_0 \in \mathbb{R}^d$,

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \gamma\frac{\mathbf{U}_k - \overline{\mathbf{X}}_k}{\rho_k} + \gamma\frac{\mathrm{D}_\lambda(\mathbf{U}_k) - \mathbf{U}_k}{\lambda} + \sqrt{2\gamma}\mathbf{Z}_{k+1}$$

$$\rho_{k+1} = \max\left(\rho_k + \delta_{k+1}\nabla_\rho \log p(\overline{\mathbf{X}}_{k+1}, \mathbf{U}_{k+1}|\boldsymbol{y}, \rho_k), 0\right)$$

where $\overline{\mathbf{X}}_{k+1} = \rho_k^{-1}\left(\sigma^{-2}\mathbf{A}^\top\boldsymbol{y} + \mathbf{U}_{k+1}\right) = \mathbb{E}_{\mathbf{X}|\mathbf{U}_{k+1},\boldsymbol{y},\rho_k}\mathbf{X}$

- **Plug-and-Play** image restoration methodology to estimate $\boldsymbol{x}^\star$ from observations

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

  - particularly suited to **Gaussian** likelihood,
  - leveraging the foundational **Denoising Diffusion Probabilistic Model**,
  - within an empirical Bayesian framework for **parameter tuning**,

to provide a **Mean A Posteriori estimator of $\boldsymbol{x}|\boldsymbol{y}$**.

- **Plug-and-Play** image restoration methodology to estimate $\boldsymbol{x}^\star$ from observations

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^\star + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

  - particularly suited to **Gaussian** likelihood,
  - leveraging the foundational **Denoising Diffusion Probabilistic Model**,
  - within an empirical Bayesian framework for **parameter tuning**,

  to provide a **Mean A Posteriori estimator of $\boldsymbol{x}|\boldsymbol{y}$**.

- Demonstrated performance on deblurring, inpainting and super-resolution
  - high PSNR and perceptual metrics performance,
  - competitive computational cost.

Limitations and perspectives

- No **convergence guarantee** when combining PnP sampling and empirical Bayes.

Limitations and perspectives

- No **convergence guarantee** when combining PnP sampling and empirical Bayes.

- Choice of noise level λ is key, but for the moment done by **cross-validation**

  joint marginal likelihood maximization over (ρ, λ).

Limitations and perspectives

- No **convergence guarantee** when combining PnP sampling and empirical Bayes.

- Choice of noise level λ is key, but for the moment done by **cross-validation**

  joint marginal likelihood maximization over (ρ, λ).

- MMSE estimator known for **missing fine details** in the posterior $\boldsymbol{x}|\boldsymbol{y}$

  Bayesian estimators aligned with **perceptual** criteria.

Limitations and perspectives

- No **convergence guarantee** when combining PnP sampling and empirical Bayes.

- Choice of noise level λ is key, but for the moment done by **cross-validation**

  joint marginal likelihood maximization over (ρ, λ).

- MMSE estimator known for **missing fine details** in the posterior $\boldsymbol{x}|\boldsymbol{y}$

  Bayesian estimators aligned with **perceptual** criteria.

- Deformation operator $\mathbf{A}$ and noise level σ^2 assumed **known**

  blind or **semi-blind** restoration problems.

Limitations and perspectives

- No **convergence guarantee** when combining PnP sampling and empirical Bayes.
- Choice of noise level λ is key, but for the moment done by **cross-validation**

  joint marginal likelihood maximization over (ρ, λ).
- MMSE estimator known for **missing fine details** in the posterior $\boldsymbol{x}|\boldsymbol{y}$

  Bayesian estimators aligned with **perceptual** criteria.
- Deformation operator **A** and noise level σ^2 assumed **known**

  blind or **semi-blind** restoration problems.
- Only independent identically distributed **Gaussian** noise

  generalization to **Poisson** or other **low-photon** noise.