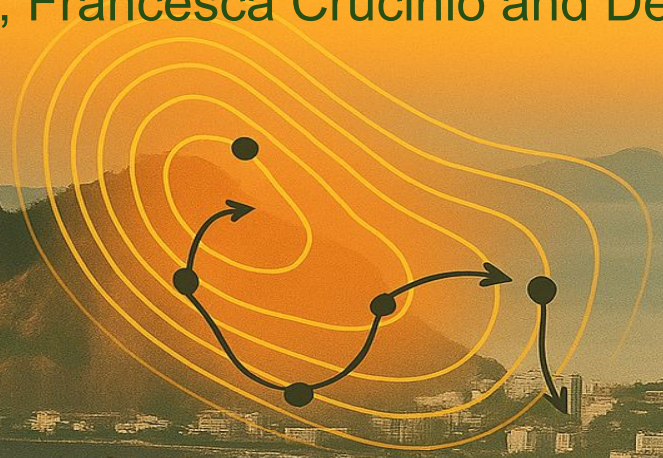
# Proximal Interacting Particle Langevin Algorithms

Paula Cordero-Encinar, Francesca Crucinio and Deniz Akyildiz

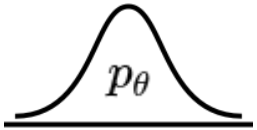


IMPERIAL

uai2025

# Some motivation: Latent Variable Models in Biology

$\theta$ : model parameters




$x$ : Genes, latent variables

$y$ : fenotype observed data

# Objectives

Perform **inference and learning in latent variable models whose joint probability distribution $p_\theta(x, y)$ is non-differentiable.**

- $\theta$ set of static parameters
- $x$ latent (unobserved, hidden, or missing) variables
- $y$ (fixed) observed data

The **statistical estimation tasks** we focus are:

- **Inference**: estimating the latent variables given the observed data and the model parameters through the computation of the posterior distribution $p_\theta(x|y)$
- **Learning**: estimating the model parameters $\theta$ given the observed data through the computation and maximisation of the marginal likelihood $p_\theta(y)$ (often intractable)

$$\mathrm{MMLE} = \bar{\theta}_\star \in \arg\max_{\theta \in \Theta} p_\theta(y) = \arg\max_{\theta \in \Theta} \int p_\theta(x, y) \mathrm{d}x$$

# Some motivation: Latent Variable Models and EM algorithm

The MMLE task in LVMs is classically solved via the Expectation-Maximisation (EM) algorithm.

- **E-step**: given $\theta_{k-1}$ we estimate the latent variables and compute $Q(\theta, \theta_{k-1}) = \mathbb{E}_{p_{\theta_{k-1}}(x|y)}[\log p_\theta(x, y)]$
- **M-step**: maximises the expectation of the E-step to provide a new estimate of $\theta$ : $\theta_k \in \arg\max_\theta Q(\theta, \theta_{k-1})$

**Challenges**

- The E and M steps are typically intractable and require approximations, which can degrade performance.
- The inherently sequential nature of EM's iterative steps limits opportunities for parallelism, making it computationally inefficient for large-scale problems.

# Background: Langevin Algorithms

Langevin algorithms are used to draw samples from a probability distribution $p(x) \propto e^{-U(x)}$ by running the following SDE

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

Under mild assumptions, this SDE has a strong solution and $p(x) \propto e^{-U(x)}$ is the unique invariant distribution of the semigroup associated with the SDE.

Langevin algorithms can be reformulated as a minimisation problem in the space of probability distributions $\min_{\mathcal{P}_2(\mathbb{R}^d)} KL(\cdot, p(x))$

# Background: Reformulating MMLE via Particle Systems

**EM algorithm is equivalent to performing coordinate descent of a free energy functional** [2], whose minimum is the maximum likelihood estimate of the latent variable model and the optimal posterior

Based on this observation, we can construct an extended stochastic dynamical system [1,2] which can be run in the space $\mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$, with the aim of jointly solving the problem of latent variable sampling and parameters optimisation. In particular, IPLA [1]

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^{N} \nabla_\theta U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N},$$
$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}, \qquad i = 1, \ldots, N.$$

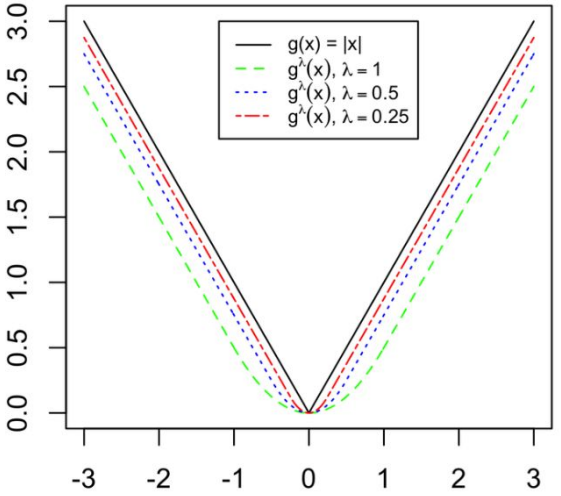
[1] Akyildiz et al. (2025) Interacting particle Langevin algorithm for maximum marginal likelihood estimation
[2] Kuntz et al. (2023) Particle algorithms for maximum likelihood training of latent variable models

# Background: Proximal map and Moreau-Yosida approximation

The $\lambda$-**proximity map or proximal operator** function of $U$ is defined for any $\lambda > 0$ as

$$\text{prox}_U^\lambda(x) \coloneqq \underset{z \in \mathbb{R}^d}{\arg\min} \left\{ U(z) + \|z - x\|^2/(2\lambda) \right\}.$$

The proximity operator $x \mapsto \text{prox}_U^\lambda(x)$ behaves similarly to a gradient mapping and moves points in the direction of the minimisers of $U$. When $U$ **is differentiable, prox corresponds to the implicit gradient step**.




Define the $\lambda$-**Moreau-Yosida approximation** of $U$ as

$$U^\lambda(x) \coloneqq \min_{z \in \mathbb{R}^d} \left\{ U(z) + \|z - x\|^2/(2\lambda) \right\}$$

# Algorithms

Our goal is to extend interacting particle algorithms for the MMLE problem to cases where the distribution $p_\theta(x, y) \propto e^{-U(\theta,x)}$ may be non-differentiable.

Our algorithms are based on discretisations of the following continuous-time interacting SDEs

$$\text{(1)} \qquad \mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^{N} \nabla_\theta U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N},$$

$$\text{(2)} \qquad \mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}.$$

Let $(\boldsymbol{\theta}_t^N)_{t\geq 0}$ be the $\theta$-marginal of the solution to the SDEs and $(\theta_n^N)_{n\in\mathbb{N}}$ be the $\theta$ iterates of any algorithm which is a discretisation of (1)–(2). Denote the $\theta$-marginal of the target measure of (1)–(2) by $\pi_{\lambda,\Theta}^N$,

$$\pi_{\lambda,\Theta}^N(\theta) \propto \int_{dx} \dots \int_{dx} e^{-\sum_{i=1}^N U^\lambda(\theta,x_i)}\mathrm{d}x_1\mathrm{d}x_2\dots\mathrm{d}x_N = \left(\int_{dx} e^{-U^\lambda(\theta,x)}\mathrm{d}x\right)^N.$$

$\pi_{\lambda,\Theta}^N$ concentrates around the maximiser of the MY approximation of the marginal likelihood as $N \to \infty$.

# Moreau-Yosida Interacting Particle Langevin Algorithm

Discretise (1)–(2) by considering $U^\lambda = g_1 + g_2^\lambda$, to derive MYIPLA:

$$\theta_{n+1}^N = \Big(1 - \frac{\gamma}{\lambda}\Big)\theta_n^N + \frac{\gamma}{N}\sum_{i=1}^N \Big( - \nabla_\theta g_1(\theta_n^N, X_n^{i,N}) + \frac{1}{\lambda}\,\mathsf{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_\theta \Big) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N},$$

$$X_{n+1}^{i,N} = \Big(1 - \frac{\gamma}{\lambda}\Big)X_n^{i,N} - \gamma\nabla_x g_1(\theta_n^N, X_n^{i,N}) + \frac{\gamma}{\lambda}\,\mathsf{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_x + \sqrt{2\gamma}\,\xi_{n+1}^{i,N}.$$

To obtain an upper bound on the distance between the iterates of our algorithm and the MMLE $\bar{\theta}_\star$

$$\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} = W_2(\delta_{\bar{\theta}_\star}, \mathcal{L}(\theta_n^N)) \leq \underbrace{W_2(\delta_{\bar{\theta}_\star}, \pi_{\lambda,\Theta}^N)}_{\text{concentration}} + \underbrace{W_2(\pi_{\lambda,\Theta}^N, \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N))}_{\text{convergence}} + \underbrace{W_2(\mathcal{L}(\theta_n^N), \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N))}_{\text{discretisation}}.$$

The concentration term can be decomposed as $W_2(\delta_{\bar{\theta}_\star}, \pi_{\lambda,\Theta}^N) \leq \|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| + W_2(\delta_{\bar{\theta}_{\star,\lambda}}, \pi_{\lambda,\Theta}^N)$, where the first term quantifies the distance between maximisers of $p_\theta(y)$ and $p_\theta^\lambda(y)$.

# Proximal Interacting Particle Gradient Langevin Algorithm

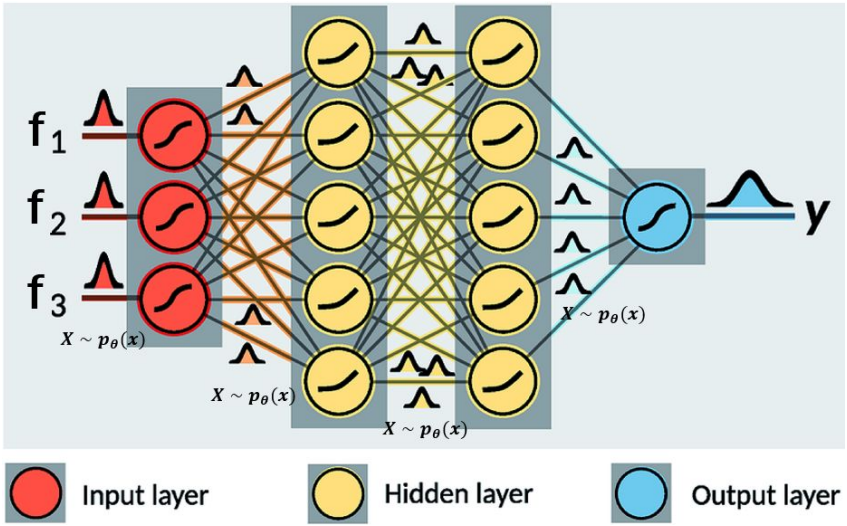
Employ a splitting scheme to discretise (1)–(2) and obtain PIPGLA:

$$\theta^N_{n+1/2} = \theta^N_n - \frac{\gamma}{N}\sum_{i=1}^N \nabla_\theta g_1(\theta^N_n, X^{i,N}_n) + \sqrt{\frac{2\gamma}{N}}\xi^{0,N}_{n+1},$$
$$X^{i,N}_{n+1/2} = X^{i,N}_n - \gamma\nabla_x g_1(\theta^N_n, X^{i,N}_n) + \sqrt{2\gamma}\,\xi^{i,N}_{n+1},$$
$$\theta^N_{n+1} = \frac{1}{N}\sum_{i=1}^N \mathrm{prox}^\lambda_{g_2}\left(\theta^N_{n+1/2}, X^{i,N}_{n+1/2}\right)_\theta,$$
$$X^{i,N}_{n+1} = \mathrm{prox}^\lambda_{g_2}\left(\theta^N_{n+1/2}, X^{i,N}_{n+1/2}\right)_x.$$

We can split the errors as follows

$$\mathbb{E}[\|\theta^N_n - \bar\theta_\star\|^2]^{1/2} = W_2(\delta_{\bar\theta_\star}, \mathcal{L}(\theta^N_n)) \le \underbrace{W_2(\delta_{\bar\theta_\star}, \pi^N_\Theta)}_{\text{concentration}} + \underbrace{W_2(\pi^N_\Theta, \mathcal{L}(\theta^N_n))}_{\text{convergence + discretisation}}.$$

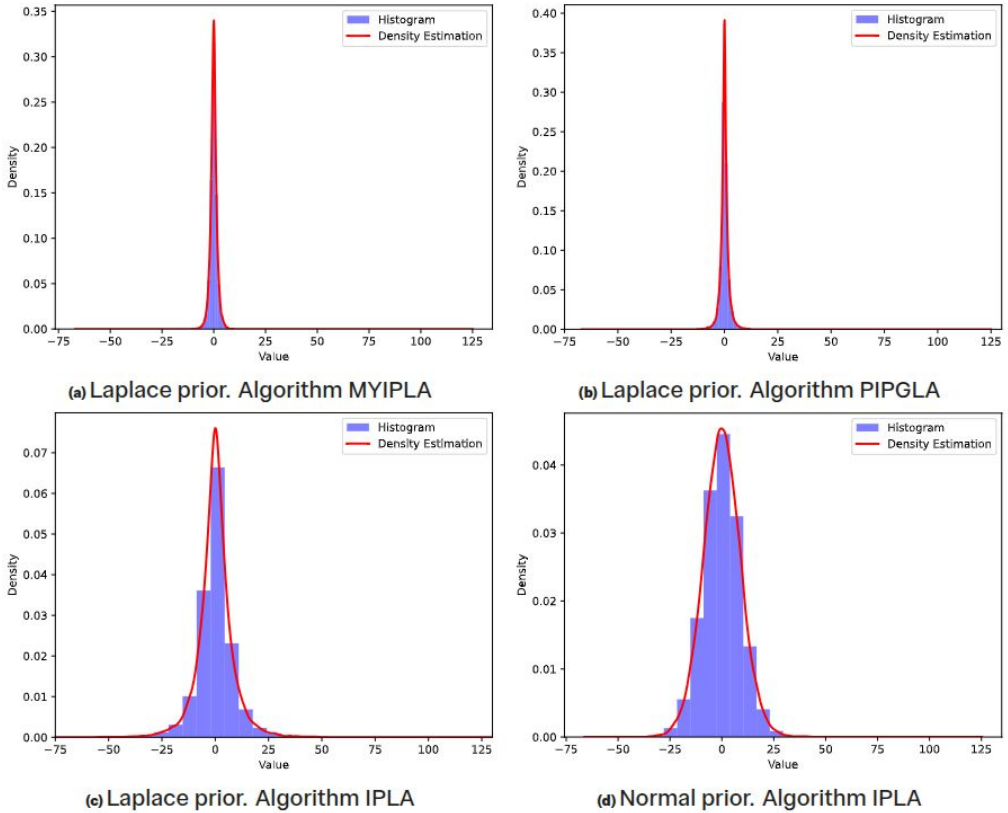
# Example I: Bayesian Neural Network with Sparse Prior




Apply a Bayesian 2-layer neural network to classify MNIST digits.

We consider a Laplace prior on the weights **x** which is a sparsity-inducing prior.

# Example I: Bayesian Neural Network with Sparse Prior




The **sparse representation of our experiment has the potential advantage of producing models that are smaller in terms of memory usage** when small weights are zeroed out.

Figure: Histogram and density estimation of the weights of a BNN for a randomly chosen particle from the final cloud of particles.

# Example II: Image Deblurring with Total Variation Prior

Recover a high-quality image from a blurred and noisy observation $y = Hx + \varepsilon$, where $H$ is a circulant blurring matrix and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$

Inverse problem is ill-conditioned $\Longrightarrow$ incorporate prior knowledge.
We use a total variation prior $e^{\theta TV(x)}$, which promotes smoothness and preserves edges

The strength of this prior depends on a hyperparameter $\theta$ that typically requires manual tuning (expert knowledge). Instead of fixing this parameter manually, we estimate its optimal value

# Example II: Image Deblurring with Total Variation Prior

(a) Original (b) Blurred (c) MYIPLA

The strength of this prior depends on a hyperparameter $\theta$ that usually requires manual tuning. **Instead, we estimate its optimal value.**

# Conclusions

Our algorithms present a **novel approach for handling Bayesian models arising from different types of non-differentiable regularisations,** including Lasso, elastic net, nuclear-norm and total variation norm.

We establish **theoretical guarantees under strong convexity** assumptions, however, in practice, **our methods perform well under more general conditions and demonstrate robustness** and stability across a range of regularisation parameter values.

**See you at the poster presentation!**


HAPPY TO CHAT MORE AT THE POSTER
PRESENTATION THIS AFTERNOON!

Thank you