Proximal Interacting Particle Langevin Algorithms

Paula Cordero Encinar[1] **Francesca R. Crucinio**[2] **O. Deniz Akyildiz**[1]

[1]Imperial College London, UK
[2]ESOMAS, University of Turin, & Collegio Carlo Alberto, Italy

Abstract

We introduce a class of algorithms, termed proximal interacting particle Langevin algorithms (PIPLA), for inference and learning in latent variable models whose joint probability density is non-differentiable. Leveraging proximal Markov chain Monte Carlo techniques and interacting particle Langevin algorithms, we propose three algorithms tailored to the problem of estimating parameters in a non-differentiable statistical model. We prove nonasymptotic bounds for the parameter estimates produced by the different algorithms in the strongly log-concave setting and provide comprehensive numerical experiments on various models to demonstrate the effectiveness of the proposed methods. In particular, we demonstrate the utility of our family of algorithms for sparse Bayesian logistic regression, training of sparse Bayesian neural networks or neural networks with non-differentiable activation functions, image deblurring, and sparse matrix completion. Our theory and experiments together show that PIPLA family can be the de facto choice for parameter estimation problems in non-differentiable latent variable models.

1 INTRODUCTION

Latent variable models (LVMs) are a class of probabilistic models which are widely used in machine learning and computational statistics for various applications, such as image, audio, and text modelling as well as in the analysis of biological data [Bishop, 2006, Murphy, 2012]. LVMs have demonstrated great success at capturing (often interpretable) latent structure in data, which is crucial in different scientific disciplines such as psychology and social sciences [Bollen, 2002, Marsh and Hau, 2007], ecology [Ovaskainen et al., 2016], epidemiology [Chavance et al., 2010, Muthén, 1992] and climate sciences [Christensen and Sain, 2012].

An LVM can be described as compactly as a parametrised joint probability distribution $p_\theta(x, y) \propto e^{-U(\theta,x)}$, where θ is a set of static parameters, x denotes latent (unobserved, hidden, or missing) variables, and finally y denotes (fixed) observed data. Given an LVM, there are two fundamental, intertwined statistical estimation tasks that must be solved simultaneously: (i) inference, which involves estimating the latent variables given the observed data and the model parameters through the computation of the posterior distribution $p_\theta(x|y)$, and (ii) learning, which involves estimating the model parameters θ given the observed data y through the computation and maximisation of the marginal likelihood $p_\theta(y)$. The learning problem is often termed maximum marginal likelihood estimation (MMLE) and the main challenge is that $p_\theta(y)$ is often intractable.

The marginal likelihood $p_\theta(y)$ (also called the *model evidence* [Bernardo and Smith, 2009]) in an LVM can be expressed as an integral, $p_\theta(y) = \int p_\theta(x, y)\mathrm{d}x$, over the latent variables. Hence the task of learning in an LVM can be formulated as solving the following optimisation problem

$$\bar{\theta}_\star \in \arg\max_{\theta\in\Theta} p_\theta(y) = \arg\max_{\theta\in\Theta} \int p_\theta(x, y)\mathrm{d}x, \quad (1)$$

where Θ is the parameter space (which will be $\mathbb{R}^{d_\theta}$ in our setting throughout). A classical algorithm for this setting is the celebrated expectation-maximisation (EM) algorithm [Dempster et al., 1977], which was first proposed in the context of missing data. The EM algorithm is an iterative procedure consisting of two main steps. Given a parameter estimate θ_k, the expectation step (E-step) computes the expected value of the log likelihood function $\log p_\theta(x, y)$ with respect to the current conditional distribution for the latent variables given the observed data $p_{\theta_k}(x|y)$, i.e., $Q(\theta, \theta_k) = \mathbb{E}_{p_{\theta_k}(x|y)}[\log p_\theta(x, y)]$. The second step is a maximisation step (M-step) which consists of maximising the expectation computed in the E-step. The EM algorithm, when it can be implemented exactly, builds a sequence of parameter estimates $(\theta_k)_{k\in\mathbb{N}}$ where $\theta_k \in \arg\max_\theta Q(\theta, \theta_{k-1})$,

which monotonically increase the marginal likelihood, i.e., $\log p_{\theta_k}(y) \geq \log p_{\theta_{k-1}}(y)$ [Dempster et al., 1977].

The wide use of the EM algorithm is due to the fact that it can be implemented using approximations for both steps [Lange, 1995, Meng and Rubin, 1993, Wei and Tanner, 1990], leveraging significant advances in Monte Carlo methods for the E-step and numerical optimisation techniques for the M-step. In particular, in most modern statistical models in machine learning, the posterior distribution, $p_\theta(x|y)$ for fixed θ, is intractable, requiring an approximation for the E-step. One way to address this is by designing Markov chain Monte Carlo (MCMC) samplers. This approach has led to significant developments, where Markov kernels based on the unadjusted Langevin algorithm (ULA) [Durmus and Moulines, 2017, Roberts and Tweedie, 1996b] have become a widespread choice in high dimensional settings thanks to their favourable theoretical properties [Chewi et al., 2022, Dalalyan, 2017, De Bortoli et al., 2021, Durmus and Moulines, 2019, Vempala and Wibisono, 2019].

Recently, Kuntz et al. [2023] explore an alternative approach for MMLE based on Neal and Hinton [1998], where they exploit the fact that the EM algorithm is equivalent to performing coordinate descent of a free energy functional, whose minimum is the maximum likelihood estimate of the latent variable model. They propose several interacting particle algorithms to address the optimisation problem. This method has led to subsequent works [Akyildiz et al., 2025, Caprio et al., 2024, Gruffaz et al., 2024, Johnston et al., 2024, Lim et al., 2024, Sharrock et al., 2023] including ours.

Contribution. This work focuses on LVMs whose joint probability density is non-differentiable by leveraging proximal methods [Combettes and Pesquet, 2011, Parikh and Boyd, 2014]. In the classical sampling case, this setting has been considered in a significant body of works, see, for example, Atchadé et al. [2017], Bernton [2018], Chen et al. [2022], Crucinio et al. [2025], Diao et al. [2023], Durmus et al. [2018], Lee et al. [2021], Pereyra [2016], Salim and Richtarik [2020], Salim et al. [2019], due to significant applications in machine learning, most notably in the use of non-differentiable regularisers. For example, this type of model naturally arises when including sparsity-inducing penalties, such as Laplace priors for regression problems or Bayesian neural networks [Williams, 1995, Yun et al., 2019], and total variation priors in image processing [Durmus et al., 2018]. They are also relevant for non-differentiable activation functions in neural networks. Specifically, we summarise our contributions below.

- We develop the first proximal interacting particle Langevin algorithm (PIPLA) family. Similar algorithms so far are investigated in the usual differentiable setting [Akyildiz et al., 2025, Johnston et al., 2024, Kuntz et al., 2023]. We extend these methods to the non-differentiable setting via the use of proximal techniques. Specifically, we propose two main algorithms, termed Moreau-Yosida interacting particle Langevin algorithm (MYIPLA) and proximal interacting particle gradient Langevin algorithm (PIPGLA).
- We present a theoretical analysis of our methods and, for comparison, of the proximal extensions we develop for the method in Kuntz et al. [2023], which we termed Moreau-Yosida particle gradient descent (MYPGD).
- We apply our methods to a variety of examples and demonstrate that the PIPLA family is a viable option for the MMLE problem in non-differentiable settings.

This paper is organised as follows. Section 2 introduces the technical background necessary to develop our methods. In Sections 3 and 4, we present the proposed algorithms along with their theoretical analysis. Section 5 presents comprehensive numerical experiments before concluding.

Notation. We endow $\mathbb{R}^d$ with the Borel σ-field $\mathcal{B}(\mathbb{R}^d)$ with respect to the Euclidean norm $\|\cdot\|$ when d is clear from context. $\mathcal{N}(x|\mu, \Sigma)$ is the multivariate Gaussian, I is the identity matrix, and $\mathcal{U}(x|a, b)$ is a uniform distribution. $\mathcal{C}^1$ denotes the space of continuously differentiable functions. We denote by $\mathcal{P}(\mathbb{R}^d)$ the set of probability measures over $\mathcal{B}(\mathbb{R}^d)$ and endow this space with the topology of weak convergence. For all $p \geq 1$, we denote by $\mathcal{P}_p(\mathbb{R}^d)$ the set of probability measures over $\mathcal{B}(\mathbb{R}^d)$ with finite p-th moment. For any $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, we denote by $W_2(\mu, \nu)$ the 2-Wasserstein distance between μ and ν. $(\mathbf{B}_t)_{t\geq 0}$ is a d-dimensional Brownian motion, and $(\xi_n)_{n\in\mathbb{N}}$ is a sequence of i.i.d. d-dimensional standard Gaussian random variables.

In the following we adopt the convention that the solution to the continuous time solution is given by bold letters, whilst other processes (including the time discretisation) are not.

2 BACKGROUND

We present the background and setting for our analysis.

2.1 LANGEVIN DYNAMICS

At the core of our approach is the use of Langevin diffusion processes [Roberts and Tweedie, 1996b], which are widely used for building advanced sampling algorithms. The Langevin stochastic differential equation (SDE) is given by

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

where $U : \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable function. Under mild assumptions, (2) admits a strong solution, and its associated semigroup has a unique invariant distribution given by $\pi(x) \propto e^{-U(x)}$ [Pavliotis, 2014]. In most cases, solving (2) analytically is not possible, however, we can

resort to a discrete-time Euler-Maruyama approximation with step size γ which gives the following Markov chain

$$X_{n+1} = X_n - \gamma \nabla U(X_n) + \sqrt{2\gamma}\xi_{n+1}. \tag{3}$$

This algorithm, known as the unadjusted Langevin algorithm (ULA) [Durmus and Moulines, 2019], exhibits favourable properties when U is μ-strongly convex and L-smooth (i.e. $\nabla U(x)$ is L-Lipschitz). In particular, it converges exponentially fast to its biased limit π^γ and the asymptotic bias is of order $\gamma^{1/2}$ [Durmus and Moulines, 2019].

2.2 MMLE WITH LANGEVIN DYNAMICS

A recent approach to solve the MMLE problem in (1) is to build an extended stochastic dynamical system 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. Kuntz et al. [2023] first proposed a method termed *particle gradient descent* (PGD) which builds on the observation made in Neal and Hinton [1998], that the EM algorithm can be expressed as a minimisation problem of the free-energy objective (see Appendix B.1 for details). By constructing a gradient flow with respect to this functional, Kuntz et al. [2023] propose the following system of SDEs

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t, \tag{4}$$

$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}, \quad i = 1, \dots, N,$$

where $U(\theta, x) = -\log p_\theta(x, y)$ given fixed observations y. By introducing a noise term in the dynamics of θ (4), which may facilitate escaping local minima in non-convex settings, Akyildiz et al. [2025] propose an interacting Langevin SDE:

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N}, \tag{5}$$

$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}, \quad i = 1, \dots, N.$$

An Euler-Maruyama discretisation of (5), provides the interacting particle Langevin algorithm (IPLA). Under strong-convexity and smoothness of U, IPLA and PGD exhibit favourable convergence properties [Akyildiz et al., 2025, Caprio et al., 2024].

2.3 PROXIMAL METHODS

Proximal methods [Combettes and Pesquet, 2011, Parikh and Boyd, 2014] use proximity mappings of convex functions, instead of gradient mappings, to construct fixed point schemes and compute function minima. We now introduce some important definitions. Consider $U : \mathbb{R}^d \to \mathbb{R}$.

Definition 1 (Proximity mappings)**.** *The λ-proximity mapping or proximal operator function of U is defined for any $\lambda > 0$ as*

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

Intuitively, the proximity operator behaves similarly to a gradient map by moving points in the direction of the minimisers of U. In fact, when U is differentiable, the proximal mapping corresponds to the implicit gradient step, as opposed to the explicit gradient step, which is known to be more stable [Parikh and Boyd, 2014]. Note that as $\lambda \to 0$, the proximity operator converges to the identity operator, while as $\lambda \to \infty$, the proximity operator maps all points to the set of minimisers of U.

The idea of proximal methods is to approximate the non-differentiable target density $\pi \propto e^{-U}$, where U is a convex lower semi-continuous function, by substituting the potential U with a smooth approximation U^λ, where the level of smoothness is controlled by the parameter $\lambda > 0$. The proximity map allows us to define a family of approximations to π, indexed by λ, and referred to as Moreau-Yosida approximation [Moreau, 1965].

Definition 2 (λ-Moreau-Yosida approximation)**.** *For any $\lambda > 0$, define the λ-Moreau-Yosida approximation of U as*

$$\begin{aligned} U^\lambda(x) &\coloneqq \min_{z\in\mathbb{R}^d}\left\{U(z) + \|z - x\|^2/(2\lambda)\right\} \\ &= U(\mathrm{prox}_U^\lambda(x)) + \|\mathrm{prox}_U^\lambda(x) - x\|^2/(2\lambda). \end{aligned}$$

Consequently, we define the λ-Moreau-Yosida approximation of π as the following density $\pi_\lambda(x) \propto e^{-U^\lambda(x)}$.

The approximation π_λ converges to π as $\lambda \to 0$ [Durmus et al., 2018, Rockafellar and Wets, 2009] and is differentiable even if π is not, with log-gradient

$$\nabla \log \pi_\lambda(x) = -\nabla U^\lambda(x) = (\mathrm{prox}_U^\lambda(x) - x)/\lambda. \tag{6}$$

Since π_λ is continuously differentiable, proximal MCMC methods [Durmus et al., 2018, Pereyra, 2016] rely on discretisations of the Langevin diffusion associated with π_λ, given by replacing the drift term in Eq. (2) with ∇U^λ, to approximately sample from π. We consider two classes of proximal Langevin algorithms, based on different discretisation schemes: Euler-Maruyama discretisations Durmus et al. [2018], Pereyra [2016], and splitting schemes Durmus et al. [2019], Ehrhardt et al. [2024], Habring et al. [2024], Klatzer et al. [2024], Salim et al. [2019].

2.3.1 Proximal Langevin methods

When $U(x) = -\log \pi(x)$ can be expressed as $U(x) = g_1(x) + g_2(x)$, with g_1, g_2 convex lower bounded functions, g_1 differentiable and g_2 proper and lower semi-continuous, Durmus et al. [2018] consider $\pi_\lambda \propto e^{-U^\lambda}$ with $U^\lambda(x) = g_1(x) + g_2^\lambda(x)$, and the corresponding Langevin diffusion

$$\mathrm{d}\mathbf{X}_{\lambda,t} = -(\nabla g_1(\mathbf{X}_{\lambda,t}) + \nabla g_2^\lambda(\mathbf{X}_{\lambda,t}))\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t,\ t \geq 0.$$

An Euler-Maruyama discretisation with step size $\gamma > 0$ results in the Moreau-Yosida ULA (MYULA) algorithm.

2.3.2 Proximal gradient MCMC methods

Inspired by the proximal gradient algorithm (see, e.g., Parikh and Boyd [2014, Section 4.2] or Combettes and Pesquet [2011]), which is a forward-backward splitting optimisation algorithm, Salim et al. [2019] propose a sampling algorithm for the case $U^\lambda(x) = g_1(x) + g_2^\lambda(x)$ termed proximal gradient Langevin algorithm (PGLA). The method consists of a forward step in the direction of ∇g_1 with an additional stochastic term, followed by a backward step using the proximity map of g_2:

$$X_{n+1/2} = X_n - \gamma \nabla g_1(X_n) + \sqrt{2\gamma}\xi_{n+1}$$
$$X_{n+1} = \mathrm{prox}_{g_2}^\lambda \left(X_{n+1/2}\right),$$

These algorithms were originally proposed as an alternative to MYULA to deal with cases in which U is the sum of a differentiable likelihood g_1 and a compactly supported g_2, since the application of the proximity map after the addition of the stochastic term guarantees that X_{n+1} remains in the support of g_2 [Salim and Richtarik, 2020]. In addition, Ehrhardt et al. [2024] give conditions under which using an approximate proximity map does not affect numerical results and generalise existing convergence bounds.

3 PROXIMAL INTERACTING PARTICLE METHODS FOR MMLE

Our goal is to extend interacting particle algorithms for the MMLE problem (1) to cases where the distribution $p_\theta(x, y) = \pi(\theta, x) \propto e^{-U(\theta,x)}$ may be non-differentiable. To achieve this, we build on the previously presented methodology, approximating the target distribution $\pi \propto e^{-U} = e^{-g_1 - g_2}$ with a Moreau-Yosida envelope π_λ and deriving a numerical scheme using either an Euler-Maruyama discretisation or a splitting scheme. In particular, we introduce three classes of proximal algorithms.

Recall that $U(\theta, x) = -\log p_\theta(x, y)$ where y is fixed. The proximal map in the MMLE setting is given below.

Remark 1. *In our scenario, the* $\arg\min$ *in the proximal map is taken over both variables* (θ, x)*, that is,*

$$\mathrm{prox}_U^\lambda(\theta, x) = (\mathrm{prox}_U^\lambda(\theta, x)_\theta, \mathrm{prox}_U^\lambda(\theta, x)_x)$$
$$= \underset{z_0 \in \mathbb{R}^{d_\theta}, z \in \mathbb{R}^{d_x}}{\arg\min} \left\{U(z_0, z) + \|(z_0, z) - (\theta, x)\|^2/(2\lambda)\right\}.$$

3.1 PROXIMAL INTERACTING PARTICLE ALGORITHMS

Our algorithms are based on different discretisation schemes of the following continuous-time interacting SDEs:

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{\frac{2}{N}}\mathrm{d}\mathbf{B}_t^{0,N}, \quad (7)$$

$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}, \quad (8)$$

where $U^\lambda = g_1 + g_2^\lambda$ is the Moreau-Yosida approximation of U. As in the case of the interacting SDE in Eq. (5), we show that (7)–(8) converges to an SDE of the McKean–Vlasov type (MKVSDE) as $N \to \infty$. In particular, if the potential U is regular enough, the MKVSDE that (7)–(8) approximates becomes arbitrarily close to that approximated by (5) as $\lambda \to 0$ (see Appendix C for a proof).

3.1.1 Moreau-Yosida Interacting Particle Langevin Algorithm (MYIPLA)

If we consider $U^\lambda = g_1 + g_2^\lambda$ as in Durmus et al. [2018], and substitute its gradient in the Euler-Maruyama discretisation of (7)–(8) we derive *MYIPLA* (Moreau-Yosida Interacting Particle Langevin algorithm):

$$\theta_{n+1}^N = \Big(1 - \frac{\gamma}{\lambda}\Big)\theta_n^N + \frac{\gamma}{N}\sum_{i=1}^N \Big(- \nabla_\theta g_1(\theta_n^N, X_n^{i,N}) + \frac{1}{\lambda}\,\mathrm{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_\theta\Big) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N}, \quad (9)$$

$$X_{n+1}^{i,N} = \Big(1 - \frac{\gamma}{\lambda}\Big)X_n^{i,N} - \gamma\nabla_x g_1(\theta_n^N, X_n^{i,N}) + \frac{\gamma}{\lambda}\,\mathrm{prox}_{g_2}^\lambda(\theta_n^N, X_n^{i,N})_x + \sqrt{2\gamma}\,\xi_{n+1}^{i,N}, \quad (10)$$

where the notation $\mathrm{prox}_{g_2}^\lambda(\theta, X)_\theta$, $\mathrm{prox}_{g_2}^\lambda(\theta, X)_x$ refers to the θ and x component of the proximal mapping $\mathrm{prox}_{g_2}^\lambda$, as mentioned in Remark 1. Setting $\gamma = \lambda$ and $g_1 = 0$ as in Pereyra [2016], we obtain a specific case of the previous algorithm, that we refer to as *PIPULA* (proximal interacting particle ULA).

3.1.2 Proximal Interacting Particle Gradient Langevin Algorithm (PIPGLA)

Inspired by the proximal gradient method [Ehrhardt et al., 2024, Salim et al., 2019], we employ a splitting scheme to discretise (7)–(8) and obtain *PIPGLA* (Proximal Interacting Particle Gradient Langevin algorithm). In this case, we perform one ULA step for both the θ and x component using ∇g_1 followed by a backward step using $\mathrm{prox}_{g_2}^\lambda$:

$$\theta_{n+1/2}^N = \theta_n^N - \frac{\gamma}{N}\sum_{i=1}^N \nabla_\theta g_1(\theta_n^N, X_n^{i,N}) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N}, \quad (11)$$

$$X_{n+1/2}^{i,N} = X_n^{i,N} - \gamma\nabla_x g_1(\theta_n^N, X_n^{i,N}) + \sqrt{2\gamma}\,\xi_{n+1}^{i,N}, \quad (12)$$

$$\theta_{n+1}^N = \frac{1}{N}\sum_{i=1}^N \mathrm{prox}_{g_2}^\lambda\left(\theta_{n+1/2}^N, X_{n+1/2}^{i,N}\right)_\theta, \quad (13)$$

$$X_{n+1}^{i,N} = \mathrm{prox}_{g_2}^\lambda\left(\theta_{n+1/2}^N, X_{n+1/2}^{i,N}\right)_x, \quad (14)$$

Setting $\lambda = \gamma$, as is common in proximal gradient algorithms (see, e.g., Salim et al. [2019]), we obtain an algorithm similar to MYIPLA except for the fact that ∇g_2^λ is evaluated at $(\theta_{n+1/2}, X_{n+1/2}^{i,N})$ instead of $(\theta_n, X_n^{i,N})$.

Similarly to PGLA, this algorithm ensures that $X_{n+1}^{i,N}$ belongs to the support of g_2 for all i. If the parameter space Θ

is convex, then also θ^N_{n+1} belongs to the support of g_2 since θ^N_{n+1} is a convex combination of elements of Θ.

3.2 PROXIMAL PARTICLE GRADIENT DESCENT METHODS (PPGD)

All the methods above incorporate a noise term in the θ-dimension, making the system more akin to a Langevin-type system, as in Akyildiz et al. [2025]. However, we also explore the case where the noise term is removed from the dynamics of θ, similar to Kuntz et al. [2023]. By removing the noise term from the dynamics of θ in (7), we obtain the following system of SDEs

$$\begin{aligned}\mathrm{d}\boldsymbol{\theta}^N_t &= -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U^\lambda(\boldsymbol{\theta}^N_t, \mathbf{X}^{i,N}_t)\mathrm{d}t,\\ \mathrm{d}\mathbf{X}^{i,N}_t &= -\nabla_x U^\lambda(\boldsymbol{\theta}^N_t, \mathbf{X}^{i,N}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}^{i,N}_t.\end{aligned}$$

Discretising this SDE system, we obtain similar algorithms to MYIPLA and PIPULA without the term $\sqrt{2\gamma/N}\xi^{0,N}_{n+1}$ in (9), which can be seen as proximal versions of PGD. Accordingly, we term these methods as proximal PGD (PPGD) and Moreau-Yosida PGD (MYPGD), respectively. We provide a detailed description of these methods and their theoretical analysis in Appendix B.

4 NONASYMPTOTIC ANALYSIS

This section presents a theoretical analysis of the parameter estimates obtained by the PIPLA family. It is important to note that similar assumptions across algorithms enable a fair comparison of convergence rates.

4.1 ASSUMPTIONS

Let $g_1, g_2 : \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x} \to \mathbb{R}$ and $U = g_1 + g_2$.

A1. *g_1 is continuously differentiable, convex, L_{g_1}-smooth and lower bounded, and g_2 is proper, convex, lower semi-continuous and lower bounded.*

A1 guarantees that $\mathrm{prox}^\lambda_{g_2}$ is well defined and that ∇U^λ is Lipschitz in both variables with constant $L \le L_{g_1} + \lambda^{-1}$ [Durmus et al., 2018, Proposition 1].

A 2. *The initial condition $Z^N_0 = (\theta_0, N^{-1/2}X^{1,N}_0, \ldots, N^{-1/2}X^{N,N}_0)$ satisfies $\mathbb{E}[\|Z^N_0\|^2] \le H$ for $H > 0$.*

A3. *g_2 is Lipschitz with constant $\|g_2\|_{\mathrm{Lip}}$.*

A4. *g_1 is μ-strongly convex.*

Remark 2. *Let $v = (\theta, x)$ and $v' = (\theta', x')$. We have that $\nabla g_2^\lambda(v) = (v - \mathrm{prox}^\lambda_{g_2}(v))/\lambda$ and $\mathrm{prox}^\lambda_{g_2}$ is firmly non expansive [Durmus et al., 2018, Eq. (7)] which implies Lipschitzness. By the Cauchy-Schwarz inequality,*

$$\begin{aligned}&\langle v - v', \nabla g_2^\lambda(v) - \nabla g_2^\lambda(v')\rangle\\ &\ge \frac{1}{\lambda}(\|v - v'\|^2 - \|v - v'\|\|\mathrm{prox}^\lambda_{g_2}(v) - \mathrm{prox}^\lambda_{g_2}(v')\|) \ge 0.\end{aligned}$$

*Therefore, under **A4**, ∇U^λ is also μ-strongly convex.*

Remark 3. *Let $\Omega \subset \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$ denote the (nonempty) set where g_2 is twice differentiable. Theorem 25.5 in [Rockafellar, 1970], guarantees that if g_2 is a proper, convex function, then g_2 is differentiable except in a set of measure zero, i.e., Ω^{c} has measure zero. Also, by Alexandrov's Theorem [Rockafellar, 1999], g_2 is twice differentiable almost everywhere—in particular, these points form a subset of $\mathsf{dom}(\nabla g_2)$. In addition, the Hessian $\nabla^2 g_2$ (or alternatively its distributional counterpart, $D^2 g_2$, if $\nabla^2 g_2$ does not exist) is symmetric and positive definite [Alberti and Ambrosio, 1999, Proposition 7.11].*

Let $\bar\theta_\star$ be the maximiser of $p_\theta(y)$. Let $\mathsf{m}_{\bar\theta_\star}$ be the restriction of the Lebesgue measure m on $\mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$ to the set $\{\bar\theta_\star\} \times \mathbb{R}^{d_x}$, which is well defined (see, e.g., [Bogachev, 2007, Section 10.6]).

A5. *We assume that $\mathsf{m}_{\bar\theta_\star}(\Omega^{\mathsf{c}} \cap (\{\bar\theta_\star\} \times \mathbb{R}^{d_x})) = 0$. Moreover,*

$$\begin{aligned}&\mathbb{E}_X[\|\nabla_\theta U(\bar\theta_\star, X)\|] \le A,\\ &\mathbb{E}_X[\|\nabla^2_{(\theta,x)} g_2(\bar\theta_\star, X)\nabla_{(\theta,x)} g_2(\bar\theta_\star, X)\|] \le B\end{aligned}$$

where $X \sim \rho_{\bar\theta_\star}(x)$ with $\rho_{\bar\theta_\star}(x) \propto e^{-U^\lambda(\bar\theta_\star, x)}$.

4.2 THE PROOF STRATEGY

Our objective is to find the MMLE $\bar\theta_\star = \arg\max_\theta p_\theta(y)$, where $p_\theta(y) = \int e^{-U(\theta,x)}\mathrm{d}x$. Therefore, we provide an upper bound on the distance between the iterates of our algorithms and $\bar\theta_\star$, that is, $\mathbb{E}[\|\bar\theta_\star - \theta_n\|^2]^{1/2}$.

Let $(\boldsymbol{\theta}^N_t)_{t\ge 0}$ be the θ-marginal of the solution to the SDE (7)–(8) and $(\theta^N_n)_{n\in\mathbb{N}}$ be the θ iterates of any algorithm which is a discretisation of (7)–(8). Denote the θ-marginal of the target measure of (7)–(8) by $\pi^N_{\lambda,\Theta}$,

$$\pi^N_{\lambda,\Theta}(\theta) \propto \left(\int_{\mathbb{R}^{d_x}} e^{-U^\lambda(\theta,x)}\mathrm{d}x\right)^N. \tag{15}$$

Note that $\mathbb{E}[\|\bar\theta_\star - \theta^N_n\|^2]^{1/2} = W_2(\delta_{\bar\theta_\star}, \mathcal{L}(\theta^N_n))$. Applying the triangle inequality of the W_2 metric, it follows:

$$\begin{aligned}W_2(\delta_{\bar\theta_\star}, \mathcal{L}(\theta^N_n)) \le &\underbrace{W_2(\delta_{\bar\theta_\star}, \pi^N_{\lambda,\Theta})}_{\text{concentration}} + \underbrace{W_2(\pi^N_{\lambda,\Theta}, \mathcal{L}(\boldsymbol{\theta}^N_{n\gamma}))}_{\text{convergence}}\\ &+ \underbrace{W_2(\mathcal{L}(\theta^N_n), \mathcal{L}(\boldsymbol{\theta}^N_{n\gamma}))}_{\text{discretisation}}.\end{aligned} \tag{16}$$

The *concentration* term characterises the concentration of the θ-marginal of the target measure of the SDE (7)–(8),

given by (15), around the maximiser of $p_\theta(y)$, denoted by $\bar{\theta}_\star$. Handling this term is not trivial since the maximisers of $p_\theta(y)$ and of $p_\theta^\lambda(y) := \int_{\mathbb{R}^{d_x}} p_\theta^\lambda(x,y)\mathrm{d}x$ are not necessarily the same as we clarify in the next section. The *convergence* term captures the convergence of the solution of the SDE to its target measure (15). Finally, the *discretisation* term characterises the error introduced by discretising the SDE. We provide nonasymptotic results for the convergence of the PIPLA family, proofs of the results are provided in Appendices A and B.

4.3 NONASYMPTOTIC ANALYSIS OF MYIPLA

Theorem 4.1 (MYIPLA)**.** *Let **A1–A5** hold. Let θ_n^N denote the iterate* (9) *and $\bar{\theta}_\star$ be the maximiser of $p_\theta(y)$. Fix $\gamma_0 \in (0, \min\{(L_{g_1}+\lambda^{-1})^{-1}, 2\mu^{-1}\})$. Then for every $\lambda > 0$ and $\gamma \in (0, \gamma_0]$, one has*

$$\begin{aligned}\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} &\le \frac{\lambda}{\mu}\Big(\frac{\|g_2\|_{\mathrm{Lip}}^2}{2}A + B\Big) + \sqrt{\frac{d_\theta}{N\mu}} \\ &+ e^{-\mu n\gamma}\left(\mathbb{E}[\|Z_0^N - z_\star\|^2]^{1/2} + \Big(\frac{d_x N + d_\theta}{N\mu}\Big)^{1/2}\right) \\ &+ C_1(1+\sqrt{d_\theta/N + d_x})\gamma^{1/2} + \mathcal{O}(\lambda^2),\end{aligned}$$

for all $n \in \mathbb{N}$, where $z_\star = (\theta_\star, N^{-1/2}x_\star, \dots, N^{-1/2}x_\star)$ and $(\theta_\star, x_\star)$ is the minimiser of U^λ and $C_1 > 0$ is a constant independent of $t, n, N, \gamma, \lambda, d_\theta, d_x$.

See Appendix A.2 for the full proof. Below, we provide a sketch of the proof, following the error decomposition in (16). The *concentration* term can be split into two parts: (1) the distance between the MMLE of the original distribution, $\bar{\theta}_\star$, and the MMLE of the MY approximation, $\bar{\theta}_{\star,\lambda}$, and (2) the concentration of the MMLE of the MY approximation around the target regularised marginal measure $\pi_{\lambda,\Theta}^N$ provided in (15). This results in

$$W_2(\delta_{\bar{\theta}_\star}, \pi_{\lambda,\Theta}^N) \le \|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| + W_2(\delta_{\bar{\theta}_{\star,\lambda}}, \pi_{\lambda,\Theta}^N),$$

where $\bar{\theta}_{\lambda,\star}$ denotes the maximiser of $p_\theta^\lambda(y)$. We derive in Proposition A.1 a novel bound for the distance between the maximisers $\bar{\theta}_\star, \bar{\theta}_{\star,\lambda}$, given by

$$\|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| \le \frac{\lambda}{\mu}\left(\frac{\|g_2\|_{\mathrm{Lip}}^2}{2}A + B\right) + \mathcal{O}(\lambda^2),$$

with A, B given in **A5**. We observe that stronger regularisation (i.e., larger values of λ) increases the distance between the two MMLEs—intuitively, in the limit $\lambda \to \infty$, all points collapse to the minimiser of the potential U, therefore this leads to a larger difference between $\bar{\theta}_\star$ and $\bar{\theta}_{\star,\lambda}$.

Next, consider the stationary measure of the SDE (7)–(8), denoted by $\pi_{\lambda,\star}^N(\theta, x_1, \dots, x_N)$, and its θ-marginal given by $\pi_{\lambda,\Theta}^N$. Using a form of the Prékopa-Leindler inequality for strong convexity [Saumard and Wellner, 2014, Theorem 3.8], $\pi_{\lambda,\Theta}^N$ is $N\mu$-strongly log-concave and by [Altschuler and Chewi, 2023, Lemma A.8]

$$W_2(\delta_{\bar{\theta}_{\star,\lambda}}, \pi_{\lambda,\Theta}^N) \le \sqrt{\frac{d_\theta}{N\mu}}, \tag{17}$$

which concludes the bound for the *concentration* term. Besides, the *convergence* term is characterised by the exponential decay of the Wasserstein-2 distance between $\pi_{\lambda,\Theta}^N$ and the θ-marginal of the solution of the SDE $\mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N)$, that is,

$$W_2(\pi_{\lambda,\Theta}^N, \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N)) \le e^{-\mu n\gamma}\left(\mathbb{E}[\|Z_0^N - z_\star\|^2]^{1/2} + \sqrt{\frac{d_x N + d_\theta}{N\mu}}\right).$$

Finally, the *discretisation* term is bounded by

$$W_2(\mathcal{L}(\theta_n^N), \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N)) \le C_1(1+\sqrt{d_\theta/N + d_x})\gamma^{1/2}.$$

This bound is derived using a strategy similar to that used in classical Langevin methods.

4.4 NONASYMPTOTIC ANALYSIS OF PIPGLA

We introduce an extra assumption regarding the least norm element $\nabla^0 g_2$ in the subdifferential of g_2, defined in A.4.

C 1. *We assume that $\|\nabla^0 g_2(\theta, x)\|^2 \le C$ for all $\theta \in \mathbb{R}^{d_\theta}$ and $x \in \mathbb{R}^{d_x}$.*

For the convergence analysis of PIPGLA, we present a novel proof that differs from the error decomposition used for MYIPLA.

Theorem 4.2 (PIPGLA)**.** *Let **A1**, **A2**, **A4** and **C1** hold. Let θ_n^N denote the iterate* (13) *and $\bar{\theta}_\star$ be the maximiser of $p_\theta(y)$. Then for $\gamma \le 1/L_{g_1}$ and $\gamma \le \lambda \le \gamma/(1-\mu\gamma)$, the following holds*

$$\begin{aligned}\mathbb{E}\left[\|\theta_n^N - \bar{\theta}_\star\|^2\right]^{1/2} &\le \sqrt{\frac{\lambda^n(1-\gamma\mu)^n}{\gamma^n}}W_2(\mathcal{L}(Z_0^N), \pi^N) \\ &+ \sqrt{\frac{d_\theta}{N\mu}} + \sqrt{\frac{\lambda(2\gamma L_{g_1}(d_\theta + N d_x) + \lambda N C)}{N\left(1 - \lambda(1-\mu\gamma)/\gamma\right)}},\end{aligned}$$

*for all $n \in \mathbb{N}$, with Z_0^N given in **A2** and $C > 0$ given in **C1** and independent of $t, n, N, \gamma, d_\theta, d_x$.*

See Appendix A.4 for the proof. Similarly to the previous result, we can split the errors as follows

$$W_2(\mathcal{L}(\theta_n^N), \delta_{\bar{\theta}_\star}) \le W_2(\delta_{\bar{\theta}_\star}, \pi_\Theta^N) + W_2(\pi_\Theta^N, \mathcal{L}(\theta_n^N)).$$

The *concentration* term $W_2(\delta_{\bar{\theta}_\star}, \pi_\Theta^N)$ can be bounded as in (17), thanks to the strong convexity assumption **A1** and the fact that the set of non-differentiable points of the potential $U(\theta, x)$ has measure zero. For the other error term, we derive novel nonasymptotic bounds for general λ, unlike Salim

Table 1: Comparison between algorithms (Section 4.5). $\delta > 0$ is any small positive constant.

	λ	N	γ	n	Evaluations of ∇g_1 and $\text{prox}_{g_2}^{\lambda}$	Indep. 1d Gaussians
MYIPLA	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(d_\theta \varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x \varepsilon^{-2-\delta})$	$\mathcal{O}(d_\theta d_x (d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_\theta d_x^2 \varepsilon^{-4-\delta})$
PIPGLA	$\mathcal{O}(\varepsilon^2)$	$\mathcal{O}(d_\theta \varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(\log \varepsilon^2 / \log d_x)$	$\mathcal{O}(d_\theta (d_\theta + d_x)\varepsilon^{-2}\frac{\log \varepsilon^2}{\log d_x})$	$\mathcal{O}(d_\theta d_x \varepsilon^{-2}\frac{\log \varepsilon^2}{\log d_x})$
MYPGD	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(d_x \varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x \varepsilon^{-2-\delta})$	$\mathcal{O}(d_x^2 (d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_x^3 \varepsilon^{-4-\delta})$

et al. [2019] (see Corollary A.13 in App. A.4). Intuitively, $W_2(\pi_\Theta^N, \mathcal{L}(\theta_n^N))$ controls the convergence of the solution of the SDE to π_Θ^N and the error due to time discretisation. The bound for the latter term is derived without introducing the law of the solution of the SDE, $\mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N)$, in contrast to the approach taken for MYIPLA.

4.5 ALGORITHM COMPARISON

Theorems 4.1, 4.2 and B.1 (Appendix) allow us to derive complexity estimates for λ, the number of particles N, γ, and the number of steps n to achieve an error $\mathbb{E}\left[\|\theta_n^N - \bar{\theta}^\star\|^2\right]^{1/2} = \mathcal{O}(\varepsilon)$, see blue columns of Table 1. These bounds are expressed in terms of the key parameters d_θ, d_x. Details for deriving these bounds are provided in Appendix D. It is important to mention that although all algorithms yield the same complexity estimates in terms of γ, MYPGD requires more stringent assumptions on γ. Specifically, while MYIPLA (Theorem 4.1) requires $\gamma_0 < \min\{(L_{g_1} + \lambda^{-1})^{-1}, 2\mu^{-1}\}$, MYPDG (Theorem B.1) requires $\gamma_0 < (L_{g_1} + \lambda^{-1} + \mu)^{-1}$, which is strictly smaller. In contrast, PIPGLA allows for a more flexible choice of $\gamma \leq 1/L_{g_1}$, but requires stronger assumptions on λ.

We also compare the algorithms in terms of their computational requirements. In particular, we evaluate the computational cost of running each algorithm for n iterations with N particles and a time discretisation step γ, while guaranteeing an $\mathcal{O}(\varepsilon)$ error. The comparison is based on the number of component-wise evaluations of ∇g_1 and $\text{prox}_{g_2}^{\lambda}$, and independent standard 1-d Gaussians samples. These costs are summarised in the green columns of Table 1.

5 EXPERIMENTS

The code is available in `https://github.com/paulaoak/proximal-ipla`. Additional experiments for low-rank matrix completion are provided in Appendix E.5, where the goal is to recover a low-rank matrix from partially observed and noisy data.

5.1 BAYESIAN LOGISTIC REGRESSION

We consider a similar set-up to De Bortoli et al. [2021] and employ a synthetic dataset consisting of $d_y = 900$ datapoints (see Appendix E.2 for details). The latent variables are the $d_x = 50$ regression weights, to which we assign an isotropic Laplace prior $p_\theta(x) = \prod_{i=1}^{d_x} \text{Laplace}(x_i|\theta, 1)$ or a uniform prior $p_\theta(x) = \prod_{i=1}^{d_x} \mathcal{U}(x_i|-\theta, \theta)$. The likelihood is given by $p_\theta(y|x) = \prod_{j=1}^{d_y} s(v_j^T x)^{y_j} s(-v_j^T x)^{1-y_j}$, where $v_j \sim \mathcal{U}(-1,1)^{\otimes d_x}$ are a set of d_x-dimensional covariates sampled from a uniform distribution and $s(u)$ is the logistic function. The true value of θ is set randomly to $\theta = -4$ for the Laplace prior and $\theta = 1.5$ for the uniform one.

Approximations of the proximal mapping for both priors are derived in Appendices E.1.1 and E.1.3, as closed-form solutions are unavailable. We compare these approximations to an iterative approach for computing the proximal map, which is feasible only for the Laplace prior due to instabilities in the uniform case. Figure 1 shows the variance of the θ estimates produced by MYIPLA and PIPGLA computed over 100 runs for different initialisations (left) and the sequence of θ estimates for 50 particles (right) in the case of a Laplace prior. We observe that the variance of the parameter estimates decreases with rate $\mathcal{O}(1/N)$ as suggested by Theorems 4.1 and 4.2, and that the iterative algorithms have a slightly lower variance compared to their

Table 2: Performance of Bayesian logistic regression for Laplace and uniform priors.

Algorithm	Approx./Iterative	NMSE (%)		Times (s)	
		Laplace	Unif	Laplace	Unif
MYPGD	Approx	6.09 ± 0.34	$\mathbf{0.60 \pm 0.23}$	91.9 ± 4.8	109.3 ± 4.6
	Iterative	4.44 ± 1.40	–	129.7 ± 15.8	–
MYIPLA	Approx	4.42 ± 1.32	15.26 ± 4.44	$\mathbf{89.9 \pm 4.2}$	$\mathbf{97.0 \pm 4.2}$
	Iterative	4.67 ± 1.60	–	120.5 ± 10.1	–
PIPGLA	Approx	2.30 ± 0.58	6.83 ± 3.97	116.5 ± 5.5	103.1 ± 8.0
	Iterative	$\mathbf{2.02 \pm 0.54}$	–	122.9 ± 6.9	–

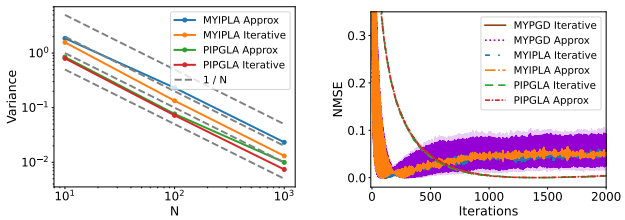
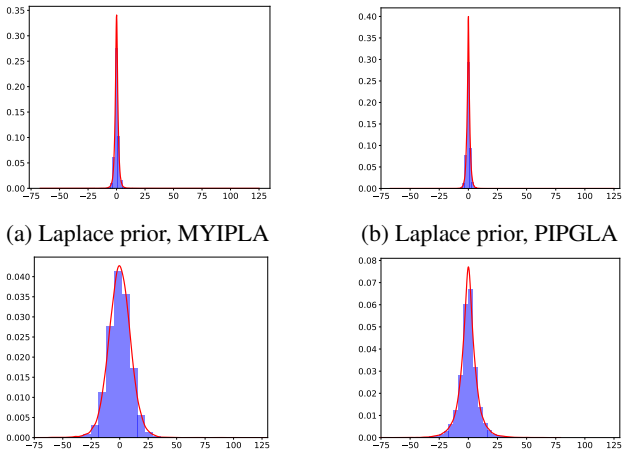


Figure 1: Laplace prior. Left: convergence rate of the variance of the parameter estimates against N produced by MYIPLA and PIPGLA over 100 runs. We see that the $\mathcal{O}(1/N)$ convergence rate holds for the second moments. Right: evolution of the normalised MSE for 50 particles over 100 runs.



(a) Laplace prior, MYIPLA (b) Laplace prior, PIPGLA

(c) Normal prior, IPLA (d) Laplace prior, IPLA

Figure 2: Histogram (blue) and density estimation (red) of the BNN weights for a randomly chosen particle. Our methods (top) produce sparser weights, which is crucial for compressibility, compared to IPLA (bottom), which ignores the non-differentiabilities.

approximate versions, with PIPGLA having better performance than MYIPLA. It is also important to highlight that for all algorithms considered, approximate solvers are on average 25% faster than iterative solvers (see Table 2). Besides, in Table 2 (expanded in Table 6), we compare the performances of the different proximal algorithms through the normalised MSEs (NMSE) for θ. In the uniform case, MYPGD outperforms all other algorithms; this is likely due to the lack of diffusive term in the corresponding SDE which is beneficial when dealing with a compactly supported prior. While PIPGLA also produces estimates within the support, they exhibit a larger bias.

5.2 BAYESIAN NEURAL NETWORK

To evaluate our algorithms on complex multimodal posteriors, we consider a Bayesian neural network with a sparsity-inducing prior on the weights for MNIST image classification. Following Yao et al. [2022] and Kuntz et al. [2023], we use a two-layer network with *tanh* activation functions and avoid the cost of computing the gradients on a big dataset by subsampling 1000 data points with labels 4 and 9. The input layer of the network has 40 nodes and 784 inputs and the output layer has 2 nodes. The latent variables are the weights, $w \in \mathbb{R}^{d_w:=40\times784}$ and $v \in \mathbb{R}^{d_v:=2\times40}$ of the input and output layers, respectively. We assign priors $p_\alpha(w) = \prod_i \text{Laplace}(w_i|0, e^{2\alpha})$ and $p_\beta(v) = \prod_i \text{Laplace}(v_i|0, e^{2\beta})$ and learn $\theta = (\alpha, \beta)$ from the data. One may ask whether the Laplace prior is more appropriate in this setting than the Normal one. Jaynes [1968] shows that the Laplace prior naturally arises for Bayesian neural network models (see Appendix E.3.1 for details). We analyse the sparsity-inducing effect of the Laplace prior by examining the distribution of the weights for a randomly chosen particle from the final particle cloud and comparing it to that obtained with a Normal prior. We note that our methods (Fig. 2a, 2b) lead to final weights with values highly concentrated around zero in comparison to the Normal prior (Fig. 2c). The sparse representation of our algorithm has the advantage of producing models that are smaller in terms of memory usage when small weights are zeroed out. This is investigated in Table 8 in the Appendix. Furthermore, we compare the performance of the PIPLA family against IPLA which ignores the non-differentiability of the model density. Figure 2d shows that, despite using a Laplace prior, IPLA fails to induce sufficient sparsity compared to our proposed methods. Quantitative results for the variance of the weights and error metrics are shown in Table 1, comparing our approach with other algorithms in the literature. Appendix E.3.2 provides additional results on more complex datasets. In particular, we apply our methods to a classification task using CIFAR10 dataset. Furthermore, in Appendix E.3.3, we also explore the application of our methods to neural networks with non-differentiable activation functions, such as ReLU.

Table 3: Bayesian neural network. Test errors and log pointwise predictive density (LPPD) achieved using the final particle cloud with $N = 50$. Computation times and standard deviation of the empirical distribution of the weight matrix w are also provided.

Algorithm	Error (%)	LPPD ($\times 10^{-1}$)	Times (s)	Std. w
MYPGD	$\mathbf{1.50 \pm 0.77}$	-1.02 ± 0.15	20	2.02
MYIPLA	2.00 ± 0.85	-1.07 ± 0.19	22	2.27
PIPGLA	2.00 ± 0.75	$\mathbf{-0.96 \pm 0.09}$	33	**1.73**
PGD	2.00 ± 0.98	-0.98 ± 0.10	**19**	8.80
SOUL	6.37 ± 1.56	-3.50 ± 2.43	55	12.09
IPLA	1.99 ± 1.01	-1.01 ± 0.25	**19**	11.70

5.3 IMAGE DEBLURRING

The objective of image deconvolution is to recover a high-quality image from a blurred and noisy observation $y = Hx + \varepsilon$, where H is a circulant blurring matrix and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. This inverse problem is ill-conditioned, a challenge that Bayesian methods address by incorporating prior knowledge. A common choice is the total variation prior, which promotes smoothness and is defined as $TV(x) = \|\nabla_d x\|_1$, where $\|\cdot\|_1$ is the ℓ_1 norm and ∇_d is the two-dimensional discrete gradient operator. However, the strength of this prior depends on a hyperparameter θ that typically requires manual tuning. Instead of fixing this parameter manually, we estimate its optimal value. Thus, the posterior distribution for the model takes the form $p_\theta(x|y) \propto C(\theta) \exp\left(-\|y - Hx\|^2/(2\sigma^2) - e^\theta TV(x)\right)$. For the experiments, we use the algorithm proposed by [Douglas and Rachford, 1956] to numerically evaluate the proximal operator of the total variation norm. Qualitative results are presented in Figure 3, with additional results provided in Appendix E.4.

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

Figure 3: Image deblurring experiment.

6 CONCLUSION

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. While our theoretical guarantees are established under strong convexity assumptions, in practice, our methods perform well under more general conditions and demonstrate robustness and stability across a range of regularisation parameter values. Moreover, unlike standard Langevin algorithms—which fail to converge for light-tailed distributions [Roberts and Tweedie, 1996a]—our proximal variants remain effective, due to the implicit regularisation introduced by the proximal map. Future work holds many promising avenues. Our theoretical framework can be extended to the non-convex setting using recent non-convex optimisation bounds [Zhang et al., 2023] and their non-differentiable adaptations. Additionally, our novel bounds on the difference between the true minimiser and the minimiser of the Moreau-Yosida approximation can also be used within recent multiscale approaches to extend them to non-differentiable settings, see, e.g., Akyildiz et al. [2024].

Acknowledgements

PCE is supported by EPSRC through the Modern Statistics and Statistical Machine Learning (StatML) CDT programme, grant no. EP/S023151/1. FRC gratefully acknowledges the "de Castro" Statistics Initative at the *Collegio Carlo Alberto* and the *Fondazione Franca e Diego de Castro*.

References

Ö Deniz Akyildiz, Michela Ottobre, and Iain Souttar. A multiscale perspective on maximum marginal likelihood estimation. *arXiv preprint arXiv:2406.04187*, 2024.

Ö. Deniz Akyildiz, Francesca Romana Crucinio, Mark Girolami, Tim Johnston, and Sotirios Sabanis. Interacting Particle Langevin Algorithm for Maximum Marginal Likelihood Estimation. *ESAIM: PROBABILITY AND STATISTICS*, 2025.

Giovanni Alberti and Luigi Ambrosio. A geometrical approach to monotone functions in $\mathbb{R}^n$. *Mathematische Zeitschrift*, 230(2):259–316, 1999.

Jason M. Altschuler and Sinho Chewi. Faster high-accuracy log-concave sampling via algorithmic warm starts. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2169–2176, 2023.

Yves F. Atchadé, Gersende Fort, and Éric Moulines. On perturbed proximal gradient algorithms. *Journal of Machine Learning Research*, 18(10):1–33, 2017.

Francis R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(35):1019–1048, 2008.

Álvaro Barbero and Suvrit Sra. Fast Newton-type Methods for Total Variation Regularization. In Lise Getoor and Tobias Scheffer, editors, *International Conference on Machine Learning*, pages 313–320, 2011.

Álvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. *Journal of Machine Learning Research*, 19(56): 1–82, 2018.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Cham, 2017.

José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

Espen Bernton. Langevin Monte Carlo and JKO splitting. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1777–1798. PMLR, 2018.

Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Vladimir I. Bogachev. *Measure Theory*, volume 2. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

Kenneth A. Bollen. Latent variables in psychology and the social sciences. *Annual Review of Psychology*, 53 (Volume 53, 2002):605–634, 2002.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Rocco Caprio, Juan Kuntz, Samuel Power, and Adam M. Johansen. Error bounds for particle gradient descent, and extensions of the log-Sobolev and Talagrand inequalities. *arXiv preprint arXiv:2403.02004*, 2024.

Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.

Michel Chavance, Sylvie Escolano, Monique Romon, Arnaud Basdevant, Blandine de Lauzon-Guillain, and Marie Aline Charles. Latent variables and structural equation models for longitudinal relationships: an illustration in nutritional epidemiology. *BMC Medical Research Methodology*, 10(1):37, 2010.

Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. Improved analysis for a proximal algorithm for sampling. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 2984–3014. PMLR, 2022.

Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of Langevin Monte Carlo from Poincaré to log-Sobolev. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1–2. PMLR, 2022.

William F. Christensen and Stephan R. Sain. Latent variable modeling for integrating output from multiple climate models. *Mathematical Geosciences*, 44(4):395–410, 2012.

Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212, 2011.

Francesca R. Crucinio, Alain Durmus, Pablo Jiménez, and Gareth O. Roberts. Optimal scaling results for Moreau-Yosida Metropolis-adjusted Langevin algorithms. *Bernoulli*, 31(3):1889–1907, 2025.

Arnak S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(3):651–676, 2017.

Valentin De Bortoli, Alain Durmus, Marcelo Pereyra, and Ana F. Vidal. Efficient stochastic optimisation by unadjusted Langevin Monte Carlo. *Statistics and Computing*, 31(3):29, 2021.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

Michael Diao, Krishnakumar Balasubramanian, Sinho Chewi, and Adil Salim. Forward-backward Gaussian variational inference via JKO in the Bures–Wasserstein space. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR, 2023.

Jim Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

Alain Durmus and Éric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin Algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017.

Alain Durmus and Éric Moulines. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

Alain Durmus, Éric Moulines, and Marcelo Pereyra. Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo: When Langevin Meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.

Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of Langevin Monte Carlo via convex optimization. *Journal of Machine Learning Research*, 20(73): 1–46, 2019.

Matthias J. Ehrhardt, Lorenz Kuger, and Carola-Bibiane Schönlieb. Proximal Langevin Sampling with Inexact Proximal Mapping. *SIAM Journal on Imaging Sciences*, 17(3):1729–1760, 2024.

Maryam Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Anupriya Gogna and Angshul Majumdar. Matrix completion incorporating auxiliary information for recommender system design. *Expert Systems with Applications*, 42(14): 5789–5799, 2015.

Jacob Vorstrup Goldman, Torben Sell, and Sumeetpal Sidhu Singh. Gradient-based Markov chain Monte Carlo for Bayesian inference with non-differentiable priors. *Journal of the American Statistical Association*, 117(540): 2182–2193, 2022.

Samuel Gruffaz, Kyurae Kim, Alain Durmus, and Jacob Gardner. Stochastic approximation with biased MCMC for expectation maximization. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 2332–2340. PMLR, 2024.

Andreas Habring, Martin Holler, and Thomas Pock. Subgradient Langevin Methods for Sampling from Nonsmooth Potentials. *SIAM Journal on Mathematics of Data Science*, 6(4):897–925, 2024.

Edwin T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, 1968.

Tim Johnston, Nikolaos Makras, and Sotirios Sabanis. Taming the Interacting Particle Langevin Algorithm – the superlinear case. *arXiv preprint arXiv:2403.19587*, 2024.

Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer New York, NY, 1991.

Teresa Klatzer, Paul Dobson, Yoann Altmann, Marcelo Pereyra, Jesus Maria Sanz-Serna, and Konstantinos C. Zygalakis. Accelerated Bayesian Imaging by Relaxed Proximal-Point Langevin Sampling. *SIAM Journal on Imaging Sciences*, 17(2):1078–1117, 2024.

Manel Kortas, Ammar Bouallegue, Tahar Ezzeddine, Vahid Meghdadi, Oussama Habachi, and Jean-Pierre Cances. Compressive sensing and matrix completion in wireless sensor networks. In *2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, pages 9–14, 2017.

Siddharth Krishna Kumar. GD doesn't make the cut: Three ways that non-differentiability affects neural network training. *arXiv preprint arXiv:2401.08426*, 2024.

Juan Kuntz, Jen Ning Lim, and Adam M. Johansen. Particle algorithms for maximum likelihood training of latent variable models. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5134–5180. PMLR, 2023.

Kenneth Lange. A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):425–437, 1995.

Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured log-concave sampling with a restricted Gaussian oracle. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2993–3050. PMLR, 15–19 Aug 2021.

Jen Ning Lim, Juan Kuntz, Samuel Power, and Adam Michael Johansen. Momentum Particle Maximum Likelihood. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29816–29871. PMLR, 2024.

Chun Sheng Liu, Bin Wang, Hong Shan, and Shan-shan Li. Survey of matrix completion models. In *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 782–787, 2018.

Lu Lu, Shin Yeonjong, Su Yanhui, and Em Karniadakis, George. Dying ReLU and Initialization: Theory and Numerical Examples. *Communications in Computational Physics*, 28(5):1671–1706, 2020.

Morteza Mardani and Georgios B. Giannakis. Estimating traffic and anomaly maps via network tomography. *IEEE/ACM Transactions on Networking*, 24:1533–1547, 2014.

Giosué Cataldo Marinó, Alessandro Petrini, Dario Malchiodi, and Marco Frasca. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing*, 520:152–170, 2023.

Herbert W. Marsh and Kit-Tai Hau. Applications of latent-variable models in educational psychology: The need for methodological-substantive synergies. *Contemporary Educational Psychology*, 32(1):151–170, 2007.

Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

J.J. Moreau. Proximité et dualité dans un espace Hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Bengt O. Muthén. Latent variable modeling in epidemiology. *Alcohol Health & Research World*, 16(4):286–292, 1992.

Radford M. Neal and Geoffrey E. Hinton. *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer Netherlands, Dordrecht, 1998.

Konstantinos Oikonomidis, Emanuel Laude, Puya Latafat, Andreas Themelis, and Panagiotis Patrinos. Adaptive Proximal Gradient Methods Are Universal Without Approximation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38663–38682. PMLR, 21–27 Jul 2024.

Otso Ovaskainen, Nerea Abrego, Panu Halme, and David Dunson. Using latent variable models to identify large networks of species-to-species associations at different spatial scales. *Methods in Ecology and Evolution*, 7(5): 549–555, 2016.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

Grigorios A Pavliotis. *Stochastic Processes and Applications. Diffusion Processes, the Fokker-Planck and Langevin Equations*, volume 60. Springer. Texts in Applied Mathematics., 2014.

Marcelo Pereyra. Proximal Markov chain Monte Carlo algorithms. *Statistics and Computing*, 26(4):745–760, 2016.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996a.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996b.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

Ralph Tyrell Rockafellar. Proximal algorithms. *Journal of Nonlinear and Convex Analysis*, 1:1–16, 1999.

Ralph Tyrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Adil Salim and Peter Richtarik. Primal dual interpretation of the proximal stochastic gradient Langevin algorithm. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3786–3796. Curran Associates, Inc., 2020.

Adil Salim, Dmitry Kovalev, and Peter Richtarik. Stochastic proximal Langevin algorithm: Potential splitting and nonasymptotic rates. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: a review. *Statistics Surveys*, 8:45, 2014.

Louis Sharrock, Daniel Dodd, and Christopher Nemeth. Tuning-Free Maximum Likelihood Training of Latent Variable Models via Coin Betting. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5134–5180. PMLR, 2023.

Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX—1989*, volume 1464 of *Lecture Notes in Mathematics*, pages 165–251. Springer, Berlin, 1991.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5): 1413–1432, 2017.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Greg CG Wei and Martin A Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

Peter M. Williams. Bayesian Regularization and Pruning Using a Laplace Prior. *Neural Computation*, 7(1):117–143, 1995.

Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for Non-mixing Bayesian Computations: The Curse and

Blessing of Multimodal Posteriors. *Journal of Machine Learning Research*, 23(79):1–45, 2022.

Jihun Yun, Peng Zheng, Eunho Yang, Aurelie Lozano, and Aleksandr Aravkin. Trimming the ℓ_1 regularizer: Statistical analysis, optimization, and applications to deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7242–7251. PMLR, 2019.

Ying Zhang, Ö Deniz Akyildiz, Theodoros Damoulas, and Sotirios Sabanis. Nonasymptotic estimates for stochastic gradient Langevin dynamics under local conditions in nonconvex optimization. *Applied Mathematics & Optimization*, 87(2):25, 2023.

Proximal Interacting Particle Langevin Algorithms (Supplementary Material)

Paula Cordero Encinar[1] **Francesca R. Crucinio**[2] **O. Deniz Akyildiz**[1]

[1]Imperial College London, UK
[2]ESOMAS, University of Turin, & Collegio Carlo Alberto, Italy

A THEORETICAL ANALYSIS OF PROXIMAL INTERACTING LANGEVIN ALGORITHMS

A.1 APPROXIMATION OF MINIMISERS

Before proceeding with the proof of Theorem 4.1 and 4.2, we derive a result controlling the distance between the maximiser of $p_\theta(y) = \int_{\mathbb{R}^{d_x}} e^{-U(\theta,x)}\mathrm{d}x$, denoted by $\bar{\theta}_\star$, and the maximiser of $p_\theta^\lambda(y) = \int_{\mathbb{R}^{d_x}} e^{-U^\lambda(\theta,x)}\mathrm{d}x$, denoted by $\bar{\theta}_{\lambda,\star}$.

For simplicity let us denote

$$k_\lambda(\theta) := \int_{\mathbb{R}^{d_x}} e^{-U^\lambda(\theta,x)}\mathrm{d}x, \qquad k(\theta) := \int_{\mathbb{R}^{d_x}} e^{-U(\theta,x)}\mathrm{d}x$$

and $K_\lambda(\theta) := -\log k_\lambda(\theta)$.

Let $\Omega \subset \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$ denote the set on which g_2 is twice differentiable. Let $\bar{\theta}_\star$ be the maximiser of k and let $\tilde{\Omega} = \Omega \cap (\{\bar{\theta}_\star\} \times \mathbb{R}^{d_x})$. By **A5**

$$k(\bar{\theta}_\star) = \int e^{-U(\bar{\theta}_\star,x)}\mathrm{d}x = \int_{\tilde{\Omega}} e^{-U(\bar{\theta}_\star,x)}\mathrm{d}x.$$

Since, k achieves a maximum at $\bar{\theta}_\star$ and g_1, g_2 are differentiable in $\tilde{\Omega}$, then

$$0 = \nabla_\theta k(\bar{\theta}_\star) = \nabla_\theta \int_{\tilde{\Omega}} e^{-U(\bar{\theta}_\star,x)}\mathrm{d}x = -\int_{\tilde{\Omega}} \Big(\nabla_\theta g_1(\bar{\theta}_\star, x) + \nabla_\theta g_2(\bar{\theta}_\star, x)\Big) e^{-U(\bar{\theta}_\star,x)}\mathrm{d}x. \tag{18}$$

Proposition A.1 (Convergence of minimisers)**.** *Under assumption **A1**, the Lipschitzness of g_2 given by **A3**, the strong convexity assumption **A4** and **A5**, it follows that*

$$\|\bar{\theta}_{\lambda,\star} - \bar{\theta}_\star\| \leq \frac{\lambda}{\mu}\Big(\frac{\|g_2\|_{\text{Lip}}^2}{2} A + B\Big) + \mathcal{O}(\lambda^2),$$

*where $\|g_2\|_{\text{Lip}}$ is the Lipschitz constant of g_2 and A, B are given in **A5**.*

Proof. To obtain a bound of $\|\bar{\theta}_{\lambda,\star} - \bar{\theta}_\star\|$ in terms of λ, we first define the measure $\pi_\lambda^1 \propto e^{-U^\lambda(\theta,x)}$ and observe that π_λ^1 is μ-strongly log-concave, since $\pi_\lambda^1 \propto e^{-U^\lambda(\theta,x)}$ and U^λ is strongly convex by **A4**. Therefore, by a form of the Prékopa-Leindler inequality for strong convexity [Saumard and Wellner, 2014, Theorem 3.8], $\pi_{\lambda,\Theta}^1 \propto e^{-K_\lambda(\theta)} = k_\lambda(\theta)$ is μ-strongly log-concave, which results in

$$\langle \bar{\theta}_{\lambda,\star} - \bar{\theta}_\star, \nabla K_\lambda(\bar{\theta}_{\lambda,\star}) - \nabla K_\lambda(\bar{\theta}_\star)\rangle \geq \mu \|\bar{\theta}_{\lambda,\star} - \bar{\theta}_\star\|^2. \tag{19}$$

Since, $\bar{\theta}_{\lambda,\star}$ is the maximiser of $k_\lambda(\theta)$ and $k_\lambda(\theta)$ is differentiable, it follows that $\nabla k_\lambda(\bar{\theta}_{\lambda,\star}) = 0$ and therefore $\nabla K_\lambda(\bar{\theta}_{\lambda,\star}) = 0$. Using the Cauchy-Schwarz inequality, we can rearrange (19) to obtain

$$\|\bar{\theta}_{\lambda,\star} - \bar{\theta}_\star\| \leq \frac{1}{\mu}\|\nabla K_\lambda(\bar{\theta}_{\lambda,\star}) - \nabla K_\lambda(\bar{\theta}_\star)\| = \frac{1}{\mu}\|\nabla K_\lambda(\bar{\theta}_\star)\| = \frac{1}{\mu k_\lambda(\bar{\theta}_\star)}\|\nabla k_\lambda(\bar{\theta}_\star)\|. \tag{20}$$

We now focus on the term $\|\nabla k_\lambda(\bar{\theta}_\star)\|$

$$\|\nabla k_\lambda(\bar{\theta}_\star)\| = \left\|\nabla_\theta \int e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\| = \left\|\int \nabla_\theta U^\lambda(\bar{\theta}_\star,x)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\|.$$

Recall that $U^\lambda(\theta,x) = g_1(\theta,x) + g_2^\lambda(\theta,x)$. For simplicity, let us assume that $\nabla_\theta g_1(\theta,x) = 0$, later we will show that this condition is not necessary. Then, we have that

$$\|\nabla k_\lambda(\bar{\theta}_\star)\| = \left\|\int \nabla_\theta g_2^\lambda(\bar{\theta}_\star,x)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\| = \left\|\int \frac{\bar{\theta}_\star - \mathrm{prox}_{g_2}^\lambda(\bar{\theta}_\star,x)_\theta}{\lambda}e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\|. \tag{21}$$

Since g_2 is convex, the problem $\min_u \left\{g_2(u) + \|v-u\|^2/(2\lambda)\right\}$ with $v = (\theta,x)$ has a unique minimum w that satisfies $\lambda\nabla g_2(w) - (v-w) = 0$. We consider the implicit system $\phi(\lambda,w) = \lambda\nabla g_2(w) - (v-w)$, and note that $\phi(0,v) = 0$ and

$$\frac{\partial\phi(\lambda,w)}{\partial w} = \lambda\nabla^2 g_2(w) + I \succ 0,$$

i.e. positive definite due to Remark 3 and assumption **A5**. Thus the Jacobian of ϕ w.r.t. w is invertible. Hence, the implicit function theorem shows that there is some locally defined ζ, such that $\zeta(0) = v$ and $\phi(\lambda,\zeta(\lambda)) = 0$. Furthermore,

$$\begin{aligned}
\frac{\partial\phi(\lambda,\zeta(\lambda))}{\partial\lambda}\bigg|_{\lambda=0} &= \left(\nabla g_2(\zeta(\lambda)) + \lambda\nabla^2 g_2(\zeta(\lambda))\frac{\partial\zeta(\lambda)}{\partial\lambda} + \frac{\partial\zeta(\lambda)}{\partial\lambda}\right)\bigg|_{\lambda=0} = 0,\\
\frac{\partial^2\phi(\lambda,\zeta(\lambda))}{\partial\lambda^2}\bigg|_{\lambda=0} &= \left(2\nabla^2 g_2(\zeta(\lambda))\frac{\partial\zeta(\lambda)}{\partial\lambda} + \lambda\frac{\partial}{\partial\lambda}\Big(\nabla^2 g_2(\zeta(\lambda))\frac{\partial\zeta(\lambda)}{\partial\lambda}\Big) + \frac{\partial^2\zeta(\lambda)}{\partial^2\lambda}\right)\bigg|_{\lambda=0} = 0,
\end{aligned}$$

which provides

$$\begin{aligned}
\frac{\partial\zeta(0)}{\partial\lambda} &= -\nabla g_2(v),\\
\frac{\partial^2\zeta(0)}{\partial\lambda^2} &= -2\nabla^2 g_2(v)\nabla g_2(v).
\end{aligned}$$

Using Taylor's expansion at $\lambda = 0$ we have

$$\mathrm{prox}_{g_2}^\lambda(v) = \zeta(\lambda) = v - \lambda\nabla g_2(v) - \lambda^2\nabla^2 g_2(v)\nabla g_2(v) + \mathcal{O}(\lambda^3). \tag{22}$$

Therefore

$$\frac{\bar{\theta}_\star - \mathrm{prox}_{g_2}^\lambda(\bar{\theta}_\star,x)_\theta}{\lambda} = \nabla_\theta g_2(\bar{\theta}_\star,x) + \lambda[\nabla^2_{(\theta,x)}g_2(\bar{\theta}_\star,x)]_{1:d_\theta}\nabla_{(\theta,x)}g_2(\bar{\theta}_\star,x) + \mathcal{O}(\lambda^2), \tag{23}$$

where the notation $[\nabla^2_{(\theta,x)}g_2(\theta,x)]_{1:d_\theta}$ means that we only take the first d_θ rows of the Hessian. For simplicity, we denote $h(\theta,x) = [\nabla^2_{(\theta,x)}g_2(\theta,x)]_{1:d_\theta}\nabla_{(\theta,x)}g_2(\theta,x)$.

Substituting (23) in (21), we have

$$\begin{aligned}
\|\nabla k_\lambda(\bar{\theta}_\star)\| &= \left\|\int_{\tilde{\Omega}} (\nabla_\theta g_2(\bar{\theta}_\star,x) + \lambda h(\bar{\theta}_\star,x) + \mathcal{O}(\lambda^2))e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\|\\
&\leq \left\|\int_{\tilde{\Omega}} \nabla_\theta g_2(\bar{\theta}_\star,x)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\| + \lambda\left(\int_{\tilde{\Omega}} \|h(\bar{\theta}_\star,X)\|\frac{e^{-U^\lambda(\bar{\theta}_\star,x)}}{k_\lambda(\bar{\theta}_\star)}\mathrm{d}x\right)k_\lambda(\bar{\theta}_\star) + \mathcal{O}(\lambda^2)k_\lambda(\bar{\theta}_\star)\\
&= \left\|\int_{\tilde{\Omega}} \nabla_\theta g_2(\bar{\theta}_\star,x)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\right\| + \lambda\mathbb{E}_X[\|h(\bar{\theta}_\star,X)\|]k_\lambda(\bar{\theta}_\star) + \mathcal{O}(\lambda^2)k_\lambda(\bar{\theta}_\star).
\end{aligned}$$

Subtracting (18) in the first term, we have

$$\begin{aligned}
\|\nabla k_\lambda(\bar{\theta}_\star)\| &\leq \Big\| \int_{\tilde{\Omega}} \nabla_\theta g_2(\bar{\theta}_\star, x)\big(e^{-U^\lambda(\bar{\theta}_\star,x)} - e^{-U(\bar{\theta}_\star,x)}\big)\mathrm{d}x\Big\| + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&\leq \int_{\tilde{\Omega}} \|\nabla_\theta g_2(\bar{\theta}_\star, x)\| e^{-U^\lambda(\bar{\theta}_\star,x)}\big(1 - e^{-U(\bar{\theta}_\star,x)+U^\lambda(\bar{\theta}_\star,x)}\big)\mathrm{d}x + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&\leq \big(1 - e^{-\lambda\|g_2\|^2_{\text{Lip}}/2}\big) \int_{\tilde{\Omega}} \|\nabla_\theta g_2(\bar{\theta}_\star, x)\| e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&= \big(1 - e^{-\lambda\|g_2\|^2_{\text{Lip}}/2}\big)\Big(\int_{\tilde{\Omega}} \|\nabla_\theta g_2(\bar{\theta}_\star, x)\| \frac{e^{-U^\lambda(\bar{\theta}_\star,x)}}{k_\lambda(\bar{\theta}_\star)}\mathrm{d}x\Big)\, k_\lambda(\bar{\theta}_\star) \\
&+ \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star)
\end{aligned}$$

where we used the fact that, since g_2 is Lipschitz by **A1**, $0 \leq U(v) - U^\lambda(v) \leq \frac{\lambda\|g_2\|^2_{\text{Lip}}}{2}$ for $v = (\theta, x)$, as shown in the proof of Durmus et al. [2018, Proposition 1].

By **A5**, we further have $\mathbb{E}_X[\|\nabla_\theta g_2(\bar{\theta}_\star, X)\|] \leq A$, $\mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] \leq B$ and thus

$$\begin{aligned}
\|\nabla k_\lambda(\bar{\theta}_\star)\| &\leq \big(1 - e^{-\lambda\|g_2\|^2_{\text{Lip}}/2}\big)\mathbb{E}_X[\|\nabla_\theta g_2(\bar{\theta}_\star, X)\|]k_\lambda(\bar{\theta}_\star) + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&= \Big(\lambda\frac{\|g_2\|^2_{\text{Lip}}}{2}\mathbb{E}_X[\|\nabla_\theta g_2(\bar{\theta}_\star, X)\|] + \lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\Big)k_\lambda(\bar{\theta}_\star) \\
&\leq \lambda\Big(\frac{\|g_2\|^2_{\text{Lip}}}{2}A + B\Big)k_\lambda(\bar{\theta}_\star) + \mathcal{O}(\lambda^2)k_\lambda(\bar{\theta}_\star). \qquad (24)
\end{aligned}$$

Putting together (20) and (24), we get that

$$\|\bar{\theta}_{\lambda,\star} - \bar{\theta}_\star\| \leq \frac{1}{\mu k_\lambda(\bar{\theta}_\star)}\|\nabla k_\lambda(\bar{\theta}_\star)\| \leq \frac{\lambda}{\mu}\Big(\frac{\|g_2\|^2_{\text{Lip}}}{2}A + B\Big) + \frac{1}{\mu}\mathcal{O}(\lambda^2) = \frac{\lambda}{\mu}\Big(\frac{\|g_2\|^2_{\text{Lip}}}{2}A + B\Big) + \mathcal{O}(\lambda^2).$$

For the case $\nabla_\theta g_1(\theta, x) \neq 0$, the same results follows since

$$\begin{aligned}
\|\nabla k_\lambda(\bar{\theta}_\star)\| &= \Big\| \int_{\tilde{\Omega}} \Big(\nabla_\theta g_1(\bar{\theta}_\star, x) + \frac{\bar{\theta}_\star - \text{prox}^\lambda_{g_2}(\bar{\theta}_\star, x)}{\lambda}\Big)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\Big\| \\
&\leq \Big\| \int_{\tilde{\Omega}} \big(\nabla_\theta g_1(\bar{\theta}_\star, x) + \nabla_\theta g_2(\bar{\theta}_\star, x)\big)e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x\Big\| + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&= \Big\| \int_{\tilde{\Omega}} \nabla_\theta U(\bar{\theta}_\star, x)\big(e^{-U^\lambda(\bar{\theta}_\star,x)} - e^{-U(\bar{\theta}_\star,x)}\big)\mathrm{d}x\Big\| + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&\leq \int_{\tilde{\Omega}} \|\nabla_\theta U(\bar{\theta}_\star, x)\| e^{-U^\lambda(\bar{\theta}_\star,x)}\big(1 - e^{-U(\bar{\theta}_\star,x)+U^\lambda(\bar{\theta}_\star,x)}\big)\mathrm{d}x \\
&+ \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&\leq \big(1 - e^{-\lambda\|g_2\|^2_{\text{Lip}}/2}\big) \int_{\tilde{\Omega}} \|\nabla_\theta U(\bar{\theta}_\star, x)\| e^{-U^\lambda(\bar{\theta}_\star,x)}\mathrm{d}x + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&= \big(1 - e^{-\lambda\|g_2\|^2_{\text{Lip}}/2}\big)\mathbb{E}_X[\|\nabla_\theta U(\bar{\theta}_\star, x)\|]k_\lambda(\bar{\theta}_\star) + \big(\lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\big)k_\lambda(\bar{\theta}_\star) \\
&= \Big(\lambda\frac{\|g_2\|^2_{\text{Lip}}}{2}\mathbb{E}_X[\|\nabla_\theta g_2(\bar{\theta}_\star, X)\|] + \lambda \mathbb{E}_X[\|h(\bar{\theta}_\star, X)\|] + \mathcal{O}(\lambda^2)\Big)k_\lambda(\bar{\theta}_\star) \\
&\leq \lambda\Big(\frac{\|g_2\|^2_{\text{Lip}}}{2}A + B\Big)k_\lambda(\bar{\theta}_\star) + \mathcal{O}(\lambda^2)k_\lambda(\bar{\theta}_\star).
\end{aligned}$$

□

A.2 MYIPLA

Following Akyildiz et al. [2025], we have the following results.

Proposition A.2. *Assuming conditions **A1** and **A2** hold, there exist a unique strong solution to* (7)–(8).

Proof. The proof follows from Karatzas and Shreve [1991] and Akyildiz et al. [2025, Proposition 1]. □

Proposition A.3 (Invariant measure)**.** *For any $N \in \mathbb{N}$, the measure $\pi^N_{\lambda,\star}(\theta, x_1, \dots, x_N) \propto \exp(-\sum_{i=1}^N U^\lambda(\theta, x_i))$ is an invariant measure for the interacting particle system* (7)-(8)*.*

Proof. The proof follows from Proposition 2 of Akyildiz et al. [2025]. □

Therefore, the system (7)-(8) has an invariant measure which admits

$$\pi^N_{\lambda,\Theta}(\theta) \propto \int_{\mathbb{R}^{d_x}} \cdots \int_{\mathbb{R}^{d_x}} e^{-\sum_{i=1}^N U^\lambda(\theta,x_i)} \mathrm{d}x_1 \dots \mathrm{d}x_N = \Big(\int_{\mathbb{R}^{d_x}} e^{-U^\lambda(\theta,x)} \mathrm{d}x \Big)^N = k_\lambda(\theta)^N,$$

as θ-marginal and can therefore act as a global optimiser of $k_\lambda(\theta)$, or more precisely of $\log k_\lambda(\theta)$. That is, let $K_\lambda(\theta) = -\log k_\lambda(\theta)$, then $\pi^N_{\lambda,\Theta}(\theta) \propto \exp(-NK_\lambda(\theta))$, concentrates around the minimiser of $K_\lambda(\theta)$, hence the maximiser of $k_\lambda(\theta)$ as $N \to \infty$. This is a classical setting in global optimisation, where N acts as the inverse temperature parameter. We now analyse the rate at which $\pi_{\lambda,\Theta}$ concentrates around the maximiser of $k(\theta)$.

Proposition A.4 (Concentration bound)**.** *Let $\pi^N_{\lambda,\Theta}$ be as defined above and $\bar\theta_\star$, $\bar\theta_{\lambda,\star}$ be the maximisers of $k(\theta)$, $k_\lambda(\theta)$, respectively. Then, under assumption **A1**, the Lipschitzness of g_2 (**A3**), the strong convexity assumption **A4** and assumption **A5**, it follows*

$$W_2(\pi^N_{\lambda,\Theta}, \delta_{\bar\theta_\star}) \le W_2(\pi^N_{\lambda,\Theta}, \delta_{\bar\theta_{\lambda,\star}}) + \|\bar\theta_{\lambda,\star} - \bar\theta_\star\| \le \sqrt{\frac{d_\theta}{\mu N}} + \frac{\lambda}{\mu}\Big(\frac{\|g_2\|^2_{\mathrm{Lip}}}{2} A + B\Big) + \mathcal{O}(\lambda^2),$$

where d_θ is the dimension of the parameter space Θ and $\|g_2\|_{\mathrm{Lip}}$ is the Lipschitz constant for g_2.

Proof. Using a form of the Prékopa-Leindler inequality for strong convexity [Saumard and Wellner, 2014, Theorem 3.8], $\pi^N_{\lambda,\Theta}$ is $N\mu$-strongly log-concave. Since it is also smooth, we can apply Lemma A.8 of Altschuler and Chewi [2023] to obtain a convergence bound for $W_2(\pi^N_{\lambda,\Theta}, \delta_{\bar\theta_{\lambda,\star}})$,

$$W_2(\pi^N_{\lambda,\Theta}, \delta_{\bar\theta_{\lambda,\star}}) \le \sqrt{\frac{d_\theta}{\mu N}}. \tag{25}$$

On the other hand, the 2-Wasserstein distance between two degenerate distributions satisfies

$$W_2(\delta_{\bar\theta_{\lambda,\star}}, \delta_{\bar\theta_\star}) = \|\bar\theta_{\lambda,\star} - \bar\theta_\star\|. \tag{26}$$

By the triangular inequality the Wasserstein distance $W_2(\pi^N_{\lambda,\Theta}, \delta_{\bar\theta_\star})$ is upper bounded by the sum of (25)-(26). We then conclude using Proposition A.1. □

The main difference with earlier works in the previous concentration bound are the second and third terms, which result from the Moreau-Yosida approximation of the non-differentiable target density π. Following on the assumptions made above and the smoothness of $\pi^N_{\lambda,\Theta}$, we have similar results to Propositions 4 and 5 of Akyildiz et al. [2025], establishing exponential ergodicity of (7)-(8) and analysing the time-discretised scheme (9)-(10).

Combining all these results, we can provide specific bounds on the accuracy of MYIPLA in terms of N, γ, n, λ and the convexity properties of U.

Theorem A.5 (Theorem 4.1 restated)**.** *Let **A1–A5** hold. Then for every λ and $\gamma_0 \in (0, \min\{(L_{g_1} + \lambda^{-1})^{-1}, 2\mu^{-1}\})$ there exist constants $C_1 > 0$ independent of $t, n, N, \gamma, \lambda, d_\theta, d_x$ such that for every $\gamma \in (0, \gamma_0]$, one has*

$$\begin{aligned}\mathbb{E}[\|\theta^N_n - \bar\theta_\star\|^2]^{1/2} \le & \sqrt{\frac{d_\theta}{N\mu}} + \frac{\lambda}{\mu}\Big(\frac{\|g_2\|^2_{\mathrm{Lip}}}{2} A + B\Big) + e^{-\mu n\gamma}\Big(\mathbb{E}[\|Z^N_0 - z_\star\|^2]^{1/2} + \Big(\frac{d_x N + d_\theta}{N\mu}\Big)^{1/2}\Big) \\ & + C_1(1 + \sqrt{d_\theta/N + d_x})\gamma^{1/2} + \mathcal{O}(\lambda^2)\end{aligned}$$

for all $n \in \mathbb{N}$, where $z_\star = (\theta_*, N^{-1/2}x_\star, \dots, N^{-1/2}x_\star)$ and $(\theta_\star, x_\star)$ is the minimiser of U^λ.

Proof. Let us denote by $\mathcal{L}(\theta_n^N)$ the θ-marginal of the measure associated to the law of MYIPLA and $\mathcal{L}(\boldsymbol{\theta}_t^N)$ the θ-marginal of the measure associated to the law of interacting particle system (7)-(8) at time t. The expectation of the norm can be decomposed into a term involving the difference between the maximisers of the marginal maximum likelihood of the Moreau-Yosida approximation of the joint density and the original density, a term concerning the concentration of $\pi^N_{\lambda,\Theta}$ around the marginal maximum likelihood estimator $\bar{\theta}_\star$, a term describing the convergence of (7)-(8) to its invariant measure, and a term involving the error induced by the time discretisation.

The first two terms are upper bounded by Proposition A.4, the third and fourth inequalities result from Proposition 4 and Proposition 5 of Akyildiz et al. [2025]. □

A.3 PIPULA

To study the theoretical guarantees of PIPULA, we observe that PIPULA is equivalent to MYIPLA when $\gamma = \lambda$ and $g_1 = 0$. We recall that in the case $g_1 = 0$, ∇U^γ is Lipschitz in both variables with constant $L \leq \gamma^{-1}$ [Durmus et al., 2018]. To obtain a similar result to Theorem 4.1 we introduce the following additional assumption.

B 1. *We assume that there exists $\mu > 0$ such that $\langle v - v', \mathrm{prox}_U^\gamma(v) - \mathrm{prox}_U^\gamma(v')\rangle \leq (1-\mu)\|v - v'\|^2$, for all $v, v' \in \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$.*

B1 implies that ∇U^γ is μ-strongly convex, i.e. $\langle v - v', \nabla U^\gamma(v) - \nabla U^\gamma(v')\rangle \geq \mu\|v - v'\|^2$ for all $v, v' \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_\theta}$. In addition, since U is a proper convex function we have that U is twice differentiable almost everywhere (see the discussion below **A5**). Let $\Omega \subset \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_x}$ denote the points where U is twice differentiable, $\bar{\theta}_\star$ be the maximiser of k and $\tilde{\Omega} = \Omega \cap (\{\bar{\theta}_\star\} \times \mathbb{R}^{d_x})$.

Using a similar strategy to that used to obtain the error bound in Theorem 4.1, we obtain the following result for PIPULA.

Corollary A.6. *Let **A1–A3**, **A5** and **B1** hold. Then for every $\gamma_0 \in (0, 2\mu^{-1})$ there exist constants $C_1 > 0$ independent of $t, n, N, \gamma, \lambda, d_\theta, d_x$ such that for every $\gamma \in (0, \gamma_0]$, one has*

$$\begin{aligned}\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} \leq & \sqrt{\frac{d_\theta}{N\mu}} + \frac{\gamma}{\mu}\Big(\frac{\|g_2\|^2_{\mathrm{Lip}}}{2}A + B\Big) \\ & + e^{-\mu n\gamma}\left(\mathbb{E}[\|Z_0^N - z_\star\|^2]^{1/2} + \Big(\frac{d_x N + d_\theta}{N\mu}\Big)^{1/2}\right) \\ & + C_1(1 + \sqrt{d_\theta/N + d_x})\gamma^{1/2} + \mathcal{O}(\gamma^2)\end{aligned}$$

for all $n \in \mathbb{N}$, where $z_\star = (\theta_*, N^{-1/2}x_\star, \ldots, N^{-1/2}x_\star)$ and $(\theta_\star, x_\star)$ is the minimiser of U^γ.

Proof. Under **A1**–**A3**, **A5** and **B1** Propositions A.1, A.2, A.3, A.4 and Proposition 4 of Akyildiz et al. [2025] hold with $\lambda = \gamma$. To obtain a bound on the discretisation error observe that under **B1** U^γ is strongly convex, since ∇U^γ is also Lipschitz continuous with constant $L \leq \gamma^{-1}$ [Durmus et al., 2018, Proposition 1], we have that ∇U^γ is co-coercive (see Theorem 1 in Gao and Pavel [2017])

$$\begin{aligned}\langle \nabla U^\gamma(x) - \nabla U^\gamma(y), x - y\rangle &\geq \frac{1}{L}\|\nabla U^\gamma(x) - \nabla U^\gamma(y)\|^2 \\ &\geq \gamma\|\nabla U^\gamma(x) - \nabla U^\gamma(y)\|^2,\end{aligned} \tag{27}$$

for every $x, y \in \mathbb{R}^d$. By plugging this result into the proof of Akyildiz et al. [2025, Lemma B.1] we obtain an equivalent result to that of Akyildiz et al. [2025, Proposition 5]:

$$W_2(\mathcal{L}(\theta_n^N), \mathcal{L}(\boldsymbol{\theta}_{n\gamma}^N)) \leq C_1(1 + \sqrt{d_\theta/N + d_x})\gamma^{1/2},$$

where $C_1 > 0$ is independent of $t, n, N, \gamma, d_\theta, d_x$ and $\gamma \in (0, \gamma_0)$ with $\gamma_0 \in (0, 2\mu^{-1})$. □

A.4 PIPGLA

Recall that PIPGLA is given by the following scheme

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

We want to prove a bound for $W_2(\mathcal{L}(\theta^N_n), \delta_{\bar\theta_\star})$, where $\mathcal{L}(\theta^N_n)$ denotes the distribution of the random variable θ^N_n. Applying the triangular inequality for Wasserstein distances

$$W_2(\mathcal{L}(\theta^N_n), \delta_{\bar\theta_\star}) \le W_2(\delta_{\bar\theta_\star}, \pi^N_\Theta) + W_2(\pi^N_\Theta, \mathcal{L}(\theta^N_n)). \tag{28}$$

The *concentration* term $W_2(\delta_{\bar\theta_\star}, \pi^N_\Theta)$ is analysed in Lemma A.14 and Theorem A.15. For the term $W_2(\pi^N_\Theta, \mathcal{L}(\theta^N_n))$, we derive a novel bound. The roadmap for obtaining this latter bound is as follows:

1. We first analyse the PGLA updates targetting a distribution $\pi_\lambda \propto \exp(-(g_1 + g_2^\lambda))$ in $\mathbb{R}^d$ given by

$$Z_n = X_n - \gamma\nabla g_1(X_n),$$
$$Y_n = Z_n + \sqrt{2\gamma}\,\xi_n,$$
$$X_{n+1} = \mathrm{prox}^\lambda_{g_2}(Y_n).$$

   We provide an error on $W_2(\mathcal{L}(X_{n+1}), \pi)$ where $\pi_\lambda \propto \exp(-(g_1 + g_2))$ for arbitrary λ and γ. One key idea for this is the use of the minimal section which quantifies the least norm element in the subdifferential set $\partial g(x)$ introduced in Definition 3.

2. We show in Corollary A.12 that the results established in 1. can be applied to a proximal gradient scheme in which the noise is scaled by $\sqrt{N}$

$$V_{n+1/2} = V_n - \gamma\nabla g_1(V_n) + \sqrt{\frac{2\gamma}{N}}\,\xi_n,$$
$$V_{n+1} = \mathrm{prox}^\lambda_{g_2}(V_{n+1/2}) = V_{n+1/2} - \lambda\nabla g_2^\lambda(V_{n+1/2}).$$

3. Taking $Z_n = (\theta^N_n, N^{-1/2}X^{1,N}_n, \ldots, N^{-1/2}X^{N,N}_n)$, PIPGLA can be expressed as

$$Z_{n+1/2} = Z_n - \gamma\nabla G_1(Z_n) + \sqrt{\frac{2\gamma}{N}}\xi_{n+1},$$
$$Z_{n+1} = \mathrm{prox}^\lambda_{G_2}(Z_{n+1/2}) = Z_{n+1/2} - \lambda\nabla G_2^\lambda(Z_{n+1/2}).$$

   where G_1 and G_2 are defined as

$$G_1(z_\theta, z_1, \ldots, z_N) = \frac{1}{N}\sum_{i=1}^N g_1(z_\theta, \sqrt{N}z_i),$$
$$G_2^\lambda(z_\theta, z_1, \ldots, z_N) = \frac{1}{N}\sum_{i=1}^N g_2^\lambda(z_\theta, \sqrt{N}z_i).$$

   Since G_1 and G_2^λ preserve the properties (strong convexity and Lipschitzness) of g_1 and g_2^λ (see Corollary A.13), we use the result in 2. to bound the error $W_2(\mathcal{L}(Z_n), \pi^N)$. Finally, using a data processing inequality it follows $W_2(\mathcal{L}(\theta^N_n), \pi^N_\Theta) \le W_2(\mathcal{L}(Z_n), \pi^N)$.

We begin by presenting some results that will be useful for proving a bound for $W_2(\pi^N_\Theta, \mathcal{L}(\theta^N_n))$.

A.4.1 Error bound for proximal gradient Langevin algorithm

We collect here a number of results adapted from Salim et al. [2019] which show convergence of the proximal gradient Langevin algorithm (PGLA) introduced in Salim et al. [2019] and recalled in Section 2.3.2. In particular, we derive convergence of the splitting scheme for general λ (which includes as a special case $\lambda = \gamma$) when both ∇g_1 and $\mathrm{prox}_{g_2}^{\lambda}$ can be computed exactly (which is a special case of the result in Salim et al. [2019] in the case $\lambda = \gamma$).

For convenience we consider the following decomposition of the PGLA update targeting a distribution $\pi_\lambda \propto \exp(-(g_1 + g_2^\lambda))$ over $\mathbb{R}^d$

$$\begin{aligned} Z_n &= X_n - \gamma \nabla g_1(X_n), \\ Y_n &= Z_n + \sqrt{2\gamma}\, \xi_n, \\ X_{n+1} &= \mathrm{prox}_{g_2}^{\lambda}(Y_n). \end{aligned}$$

We are going to derive a bound for $W_2(\mathcal{L}(X_{n+1}), \pi)$, where $\mathcal{L}(X_{n+1})$ denotes the distribution of the random variable X_{n+1}. For every π-integrable function $g : \mathbb{R}^d \to \mathbb{R}$, we define $\mathcal{E}_g(\pi) = \int g \mathrm{d}\pi$ and we denote $\mathcal{F} = \mathcal{E}_{g_1+g_2} + \mathcal{H}$, where $\mathcal{H}$ is the negative entropy $\mathcal{H}(\pi) = \int \log \pi \, \mathrm{d}\pi$. We also introduce the subdifferential of a convex function and its minimal section, since we will use them for our proofs.

Definition 3. *[Subdifferential and minimal section]* For any convex function $g : \mathbb{R}^d \to \mathbb{R}$, its subdifferential evaluated at x is the set

$$\partial g(x) := \{d \in \mathbb{R}^d \mid g(x) + \langle d, y - x\rangle \leq g(y) \forall y \in \mathbb{R}^d\}.$$

Thanks to Bauschke and Combettes [2017, Proposition 16.4], we have that $\partial g(x)$ is a nonempty closed convex set. So the projection of 0 onto $\partial g(x)$, that is, the least norm element in the set $\partial g(x)$, is well defined, and we refer to this element as $\nabla^0 g(x)$. Following Salim et al. [2019, Section 3.1], we name the function $\nabla^0 g : \mathbb{R}^d \to \mathbb{R}$ the minimal section of ∂g.

Following Salim et al. [2019], we derive our results under the following assumptions on g_1, g_2.

D 1. *We assume that $g_1 \in \mathcal{C}^1$ is convex, gradient Lipschitz with constant L_{g_1} and lower bounded, and g_2 is proper, convex, lower semi-continuous and lower bounded.*

D 2. *g_1 is μ-strongly convex.*

D 3. *Assume that $\|\nabla^0 g_2(x)\|^2 \leq C$ for every $x \in \mathbb{R}^d$.*

In particular, **D1** and **D2** are equivalent to **A1** and **A4** for the target $\pi_\lambda(\theta, x) \propto \exp(-U^\lambda(\theta, x))$. Similarly, **D3** is equivalent to **C1**.

Lemma A.7. *Let **D1** and **D2** hold and assume $\gamma \leq 1/L_{g_1}$. Then, for all $n \in \mathbb{N}$*

$$2\gamma\big[\mathcal{E}_{g_1}(\mathcal{L}(Z_n)) - \mathcal{E}_{g_1}(\pi)\big] \leq (1 - \gamma\mu) W_2^2(\mathcal{L}(X_n), \pi) - W_2^2(\mathcal{L}(Z_n), \pi).$$

Proof. Let $a \in \mathbb{R}^d$, using that g_1 is μ–strongly convex

$$\begin{aligned} \|Z_n - a\|^2 &= \|X_n - a\|^2 - 2\gamma\langle \nabla g_1(X_n), X_n - a\rangle + \gamma^2 \|\nabla g_1(X_n)\|^2 \\ &\leq \|X_n - a\|^2 + 2\gamma\big(g_1(a) - g_1(X_n) - \frac{\mu}{2}\|X_n - a\|^2\big) + \gamma^2\|\nabla g_1(X_n)\|^2 \\ &= (1 - \gamma\mu)\|X_n - a\|^2 + 2\gamma\big(g_1(a) - g_1(X_n)\big) + \gamma^2\|\nabla g_1(X_n)\|^2. \end{aligned} \tag{29}$$

Since g_1 is gradient Lipschitz with constant L_{g_1} and $Z_n - X_n = -\gamma \nabla g_1(X_n)$

$$\begin{aligned} g_1(Z_n) &\leq g_1(X_n) + \langle \nabla g_1(X_n), Z_n - X_n\rangle + \frac{L_{g_1}}{2}\|Z_n - X_n\|^2 \\ &= g_1(X_n) - \gamma\Big(1 - \frac{\gamma L_{g_1}}{2}\Big)\|\nabla g_1(X_n)\|^2 \\ &\leq g_1(X_n) - \frac{\gamma}{2}\|\nabla g_1(X_n)\|^2, \end{aligned}$$

where in the last inequality we have used that $\gamma \leq 1/L_{g_1}$. Reordering terms gives the following upper bound

$$\gamma^2 \|\nabla g_1(X_n)\|^2 \leq 2\gamma\big(g_1(X_n) - g_1(Z_n)\big).$$

Plugging this into (29) we have

$$\|Z_n - a\|^2 \leq (1-\gamma\mu)\|X_n - a\|^2 + 2\gamma\big(g_1(a) - g_1(Z_n)\big).$$

It is important to note, the above inequality is true for any a, X_n, and Z_n where $Z_n = X_n - \gamma\nabla g_1(X_n)$ (as deterministic vectors with appropriate dimension). Now, let $(a, X_n) \sim \nu(\mathrm{d}a, \mathrm{d}x_n)$ with marginal $\nu^a(\mathrm{d}a) = \pi(\mathrm{d}a)$. Taking conditional expectation w.r.t. Z_n given $\sigma(a, X_n)$, we obtain

$$\mathbb{E}\big[\|Z_n - a\|^2 \big| \sigma(a, X_n)\big] \leq (1-\gamma\mu)\|X_n - a\|^2 + 2\gamma\big(g_1(a) - \mathbb{E}\big[g_1(Z_n)|\sigma(a, X_n)\big]\big).$$

By taking the unconditional expectation (i.e. w.r.t. ν), we get

$$\mathbb{E}\big[\|Z_n - a\|^2\big] \leq (1-\gamma\mu)\mathbb{E}_\nu\big[\|X_n - a\|^2\big] + 2\gamma\big(\mathcal{E}_{g_1}(\pi) - \mathcal{E}_{g_1}(\mathcal{L}(Z_n))\big).$$

By the definition of the Wasserstein distance we get

$$W_2^2(\mathcal{L}(Z_n), \pi) \leq (1-\gamma\mu)\mathbb{E}_\nu\big[\|X_n - a\|^2\big] + 2\gamma\big(\mathcal{E}_{g_1}(\pi) - \mathcal{E}_{g_1}(\mathcal{L}(Z_n))\big).$$

Note that the last inequality is true for all ν with prescribed marginal above. In particular, we can take the infimum over all such couplings and inequality would still hold for the infimum. This leads to

$$W_2^2(\mathcal{L}(Z_n), \pi) \leq (1-\gamma\mu)W_2^2(\mathcal{L}(X_n), \pi) + 2\gamma\big(\mathcal{E}_{g_1}(\pi) - \mathcal{E}_{g_1}(\mathcal{L}(Z_n))\big),$$

which is the desired result. □

Lemma A.8. *Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex function and g^λ its λ-Moreau-Yosida approximation. Consider $a, y_0, y_1 \in \mathbb{R}^d$ such that $y_1 = \mathrm{prox}_g^\lambda(y_0)$. Then,*

$$\|y_1 - a\|^2 \leq \|y_0 - a\|^2 - 2\lambda\left(g(y_0) - g(a)\right) + \lambda^2\|\nabla g^0(y_0)\|^2.$$

Proof. Recalling that $\mathrm{prox}_g^\lambda(y_0) = y_0 - \lambda\nabla g^\lambda(y_0)$ we have

$$\|y_1 - a\|^2 = \|y_0 - a\|^2 - 2\lambda\langle\nabla g^\lambda(y_0), y_0 - a\rangle + \lambda^2\|\nabla g^\lambda(y_0)\|^2. \tag{30}$$

Using that $y_1 = y_0 - \lambda\nabla g^\lambda(y_0)$ we can write

$$\langle\nabla g^\lambda(y_0), y_0 - a\rangle = \langle\nabla g^\lambda(y_0), y_1 - a\rangle + \lambda\|\nabla g^\lambda(y_0)\|^2.$$

Since $\nabla g^\lambda(y_0)$ belongs to the subdifferential of $g(y_1)$, i.e. $\nabla g^\lambda(x) \in \partial g(\mathrm{prox}_g^\lambda(x))$ [Bauschke and Combettes, 2017, Proposition 16.44], we further have that

$$\langle\nabla g^\lambda(y_0), y_1 - a\rangle \geq g(y_1) - g(a),$$

from which we obtain

$$-2\lambda\langle\nabla g^\lambda(y_0), y_0 - a\rangle \leq -2\lambda\left(g(y_1) - g(a) + \lambda\|\nabla g^\lambda(y_0)\|^2\right).$$

Recalling the definition of Moreau-Yosida approximation in Definition 2 we have that $g^\lambda(y_0) = g(y_1) + \|y_0 - y_1\|^2/(2\lambda)$; plugging this into the equation above gives

$$\begin{aligned} -2\lambda\langle\nabla g^\lambda(y_0), y_0 - a\rangle &\leq -2\lambda\left(g^\lambda(y_0) - g(a)\right) - 2\lambda^2\|\nabla g^\lambda(y_0)\|^2 + \|y_1 - y_0\|^2 \\ &= -2\lambda(g^\lambda(y_0) - g(a)) - \lambda^2\|\nabla g^\lambda(y_0)\|^2. \end{aligned} \tag{31}$$

Finally, using Salim et al. [2019, Lemma 9] which states that $g^\lambda(x) \geq g(x) - \lambda\|\nabla^0 g(x)\|/2$, where $\nabla^0 g$ is the minimal section introduced in Definition 3, and combining (30) and (31) we have

$$\|y_1 - a\|^2 \leq \|y_0 - a\|^2 - 2\lambda\left(g^\lambda(y_0) - g(a)\right) \leq \|y_0 - a\|^2 - 2\lambda\left(g(y_0) - g(a)\right) + \lambda^2\|\nabla^0 g(x)\|^2.$$

□

Lemma A.9. *Let **D1–D3** hold. Then,*

$$2\lambda\big[\mathcal{E}_{g_2}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_2}(\pi)\big] \leq W_2^2(\mathcal{L}(Y_n), \pi) - W_2^2(\mathcal{L}(X_{n+1}), \pi) + \lambda^2 C$$

Proof. Applying Lemma A.8 with $y_0 = Y_n$, $y_1 = X_{n+1}$ and $g = g_2$, we have

$$\|X_{n+1} - a\|^2 \leq \|Y_n - a\|^2 - 2\lambda\big(g_2(Y_n) - g_2(a)\big) + \lambda^2 \|\nabla^0 g_2(Y_n)\|^2.$$

Now, let a be a random vector sampled from the distribution with density π. Taking expectations in the previous expression and using the definition of the Wasserstein distance we obtain

$$\begin{aligned} W_2^2(\mathcal{L}(X_{n+1}), \pi) &\leq \mathbb{E}\big[\|Y_n - a\|^2\big] - 2\lambda\big(\mathcal{E}_{g_2}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_2}(\pi)\big) + \lambda^2 \mathbb{E}[\|\nabla^0 g_2(Y_n)\|^2] \\ &\leq \mathbb{E}\big[\|Y_n - a\|^2\big] - 2\lambda\big(\mathcal{E}_{g_2}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_2}(\pi)\big) + \lambda^2 C. \end{aligned}$$

Finally, taking the infimum over all couplings Y_n, a of $\mathcal{L}(Y_n), \pi$, it follows that

$$W_2^2(\mathcal{L}(X_{n+1}), \pi) \leq W_2^2(\mathcal{L}(Y_n), \pi) - 2\lambda\big(\mathcal{E}_{g_2}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_2}(\pi)\big) + \lambda^2 C.$$

□

Theorem A.10. *Let **D1–D3** hold and assume that $\gamma \leq 1/L_{g_1}$. Then, for all $n \in \mathbb{N}$*

$$\begin{aligned} 2\gamma\,\mathrm{KL}(\mathcal{L}(Y_n) \mid \pi) \leq &(1 - \gamma\mu) W_2^2(\mathcal{L}(X_n), \pi) - \frac{\gamma}{\lambda} W_2^2(\mathcal{L}(X_{n+1}), \pi) \\ &- \Big(1 - \frac{\gamma}{\lambda}\Big) W_2^2(\mathcal{L}(Y_n), \pi) + \gamma(2\gamma L_{g_1} d + \lambda C). \end{aligned}$$

Proof. Since $g_1 + g_2$ is convex by assumption the following holds $\pi \in \mathcal{P}_2(\mathbb{R}^d)$, $\mathcal{H}(\pi) < \infty$, $\mathcal{E}_{g_1+g_2}(\pi) < \infty$ and for all $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ satisfying $\mathcal{E}_{g_1+g_2}(\mu) < \infty$,

$$\mathrm{KL}(\mu \mid \pi) = \mathcal{E}_{g_1+g_2}(\mu) + \mathcal{H}(\mu) - (\mathcal{E}_{g_1+g_2}(\pi) + \mathcal{H}(\pi)) = \mathcal{F}(\mu) - \mathcal{F}(\pi).$$

We can further decompose $\mathcal{E}_{g_1+g_2}(\mu) = \mathcal{E}_{g_1}(\mu) + \mathcal{E}_{g_2}(\mu)$. Using Durmus et al. [2019, Lemma 5] we have that the negative entropy satisfies the following inequality

$$2\gamma\big[\mathcal{H}(\mathcal{L}(Y_n)) - \mathcal{H}(\pi)\big] \leq W_2^2(\mathcal{L}(Z_n), \pi) - W_2^2(\mathcal{L}(Y_n), \pi). \tag{32}$$

Since g_1 is L_{g_1}-gradient Lipschitz and strongly convex, it follows that

$$0 \leq g_1(Y_n) - g_1(Z_n) + \langle \nabla g_1(Z_n), Z_n - Y_n \rangle \leq \frac{L_{g_1}}{2} \|Y_n - Z_n\|^2.$$

Note that $Y_n - Z_n = \sqrt{2\gamma}\xi_n$ is independent of Z_n, $\mathbb{E}[Y_n - Z_n] = 0$ and $\mathbb{E}[\|Y_n - Z_n\|^2] = 2\gamma d$, where d is the dimension of the standard Gaussian random variable ξ_n. Therefore, taking expectations in the previous inequality we get

$$2\gamma\big[\mathcal{E}_{g_1}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_1}(\mathcal{L}(Z_n))\big] \leq 2\gamma^2 L_{g_1} d. \tag{33}$$

On the other hand, by Lemmas A.7 and A.9 we have

$$2\gamma\big[\mathcal{E}_{g_1}(\mathcal{L}(Z_n)) - \mathcal{E}_{g_1}(\pi)\big] \leq (1 - \gamma\mu) W_2^2(\mathcal{L}(X_n), \pi) - W_2^2(\mathcal{L}(Z_n), \pi), \tag{34}$$

$$2\gamma\big[\mathcal{E}_{g_2}(\mathcal{L}(Y_n)) - \mathcal{E}_{g_2}(\pi)\big] \leq \frac{\gamma}{\lambda} W_2^2(\mathcal{L}(Y_n), \pi) - \frac{\gamma}{\lambda} W_2^2(\mathcal{L}(X_{n+1}), \pi) + \gamma\lambda C. \tag{35}$$

Summing up (32)-(35) and using that $\mathrm{KL}(\mathcal{L}(Y_n) \mid \pi) = \mathcal{F}(\mathcal{L}(Y_n)) - \mathcal{F}(\pi)$ we have the desired result.

□

Corollary A.11. *Let **D1–D3** hold and assume that $\gamma \leq 1/L_{g_1}$ and $\gamma \leq \lambda \leq \gamma/(1 - \mu\gamma)$. Then*

$$W_2^2(\mathcal{L}(X_n), \pi) \leq \frac{\lambda^n (1 - \gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(X_0), \pi) + \frac{\lambda(2\gamma L_{g_1} d + \lambda C)}{1 - \lambda(1 - \mu\gamma)/\gamma}.$$

Proof. Since the KL divergence and the Wasserstein distance are always non-negative and we assume that $\gamma \leq \lambda$, we have by Theorem A.10 that for all $n \in \mathbb{N}$

$$W_2^2(\mathcal{L}(X_{n+1}), \pi) \leq \frac{\lambda(1-\gamma\mu)}{\gamma} W_2^2(\mathcal{L}(X_n), \pi) + \lambda(2\gamma L_{g_1} d + \lambda C).$$

Unrolling this recurrence we get

$$\begin{aligned} W_2^2(\mathcal{L}(X_n), \pi) &\leq \frac{\lambda^n(1-\gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(X_0), \pi) + \lambda(2\gamma L_{g_1} d + \lambda C) \sum_{i=0}^{n-1} \frac{\lambda^i(1-\gamma\mu)^i}{\gamma^i} \\ &\leq \frac{\lambda^n(1-\gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(X_0), \pi) + \frac{\lambda(2\gamma L_{g_1} d + \lambda C)}{1 - \lambda(1-\gamma\mu)/\gamma}, \end{aligned}$$

where we have used the assumption $\lambda \leq \gamma/(1-\mu\gamma)$. □

A.4.2 Convergence and discretisation bounds

We start by showing that the results established above can be applied to a proximal gradient scheme in which the noise is scaled by $\sqrt{N}$

$$\begin{aligned} V_{n+1/2} &= V_n - \gamma \nabla g_1(V_n) + \sqrt{\frac{2\gamma}{N}}\, \xi_n, \\ V_{n+1} &= V_{n+1/2} - \lambda \nabla g_2^\lambda(V_{n+1/2}). \end{aligned} \tag{36}$$

Corollary A.12 (Rescaled noise)**.** *Let **D1–D3** hold and assume that $\gamma \leq 1/L_{g_1}$ and $\gamma \leq \lambda \leq \gamma/(1-\mu\gamma)$. Then,*

$$W_2^2(\mathcal{L}(V_n), \pi^N) \leq \frac{\lambda^n(1-\gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(V_0), \pi^N) + \frac{\lambda(2\gamma L_{g_1} d + \lambda N C)}{N\,(1 - \lambda(1-\mu\gamma)/\gamma)}.$$

Proof. Let $\tilde{g}_1 = N g_1$ and $\tilde{g}_2 = N g_2$. It is easy to check that $\tilde{g}_1$ is $(N L_{g_1})$–gradient Lipschitz and $(N\mu)$–strongly convex. In addition, we have that

$$\operatorname{prox}_{g_2}^\lambda(x) = \operatorname*{arg\,min}_{z \in \mathbb{R}^d} \frac{\tilde{g}_2(x)}{N} + \frac{\|x-z\|^2}{2\lambda} = \operatorname*{arg\,min}_{z \in \mathbb{R}^d} \frac{1}{N}\left(\tilde{g}_2(x) + \frac{\|x-z\|^2}{2\lambda/N}\right) = \operatorname{prox}_{\tilde{g}_2}^{\lambda/N}(x),$$

since the $\arg\min$ does not change if the function is multiplied by a constant, which results in $\nabla g_2^\lambda = \nabla \tilde{g}_2^{\lambda/N}/N$. Thus, (36) can be rewritten as

$$\begin{aligned} V_{n+1/2} &= V_n - \tilde{\gamma} \nabla \tilde{g}_1(V_n) + \sqrt{2\tilde{\gamma}}\, \xi_n, \\ V_{n+1} &= V_{n+1/2} - \tilde{\lambda} \nabla \tilde{g}_2^{\tilde{\lambda}}(V_{n+1/2}), \end{aligned}$$

where $\tilde{\gamma} = \gamma/N$ and $\tilde{\lambda} = \lambda/N$. Note that the subdifferential set satisfies $\partial \tilde{g}_2 = N \partial g_2$. Therefore, since $\|\nabla^0 g_2(x)\|^2 \leq C$ for all $x \in \mathbb{R}^d$ by **D3**, it follows that $\|\nabla^0 \tilde{g}_2(x)\|^2 \leq N^2 C$. Therefore, taking $\tilde{\gamma} \leq 1/(N L_{g_1})$ which is equivalent to $\gamma \leq 1/L_{g_1}$, and applying Corollary A.11 the result follows. □

In order to be able to use the bound obtained in Corollary A.12, we rewrite PIPGLA as the algorithm given in (36). To do so, define

$$\begin{aligned} G_1(z_\theta, z_1, \ldots, z_N) &= \frac{1}{N} \sum_{i=1}^N g_1(z_\theta, \sqrt{N} z_i), \\ G_2^\lambda(z_\theta, z_1, \ldots, z_N) &= \frac{1}{N} \sum_{i=1}^N g_2^\lambda(z_\theta, \sqrt{N} z_i). \end{aligned}$$

Note that the gradients of these functions are given by

$$\nabla G_1(z_\theta, z_1, \dots, z_N) = \left(N^{-1} \sum_{i=1}^N \nabla_\theta g_1(z_\theta, \sqrt{N} z_i), N^{-1/2} \nabla_{z_1} g_1(z_\theta, \sqrt{N} z_1), \dots, N^{-1/2} \nabla_{z_N} g_1(z_\theta, \sqrt{N} z_N) \right)^\intercal$$

and similarly for G_2^λ.

Taking $Z_n = (\theta_n^N, N^{-1/2} X_n^{1,N}, \dots, N^{-1/2} X_n^{N,N})$, PIPGLA can be expressed as

$$\begin{aligned} Z_{n+1/2} &= Z_n - \gamma \nabla G_1(Z_n) + \sqrt{\frac{2\gamma}{N}} \xi_{n+1}, \\ Z_{n+1} &= Z_{n+1/2} - \lambda \nabla G_2^\lambda(Z_{n+1/2}). \end{aligned} \tag{37}$$

Corollary A.13. *Let $Z_n = (\theta_n^N, N^{-1/2} X_n^{1,N}, \dots, N^{-1/2} X_n^{N,N})$ and $\pi^N \propto \exp(-N(G_1 + G_2))$. Suppose that **A1**, **A4** and **C1** hold true and assume $\gamma \le 1/L_{g_1}$ and $\gamma \le \lambda \le \gamma/(1 - \mu\gamma)$. Then,*

$$W_2^2(\mathcal{L}(Z_n), \pi^N) \le \frac{\lambda^n (1 - \gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(Z_0), \pi^N) + \frac{\lambda(2\gamma L_{g_1}(d_\theta + N d_x) + \lambda N C)}{N\,(1 - \lambda(1 - \mu\gamma)/\gamma)}.$$

Proof. Note that if **C1** holds then $\|\nabla^0 G_2(z)\|^2 \le C$ for every z. To see this note that

$$\partial G_2(z) = \partial G_2(z_\theta, z_1, \dots, z_N) = \frac{1}{N} \sum_{i=1}^N \partial g_2(z_\theta, \sqrt{N} z_i).$$

Therefore, using the fact that $(N^{-1} \sum_i a_i)^2 \le N^{-1} \sum_i a_i^2$, we get that the minimal section satisfies

$$\|\nabla^0 G_2(z)\|^2 \le \frac{1}{N} \sum_{i=1}^N \left\| \nabla^0 g_2(z_\theta, \sqrt{N} z_i) \right\|^2 \le C.$$

In addition, observe that **A1**, **A4** imply that G_1, G_2 and G_2^λ are convex since g_1, g_2 and g_2^λ are convex, and G_1 is also μ-strongly convex and L_{g_1}-gradient Lipschitz. The proof then follows from Corollary A.12. □

Before proving our final result for $W_2(\mathcal{L}(\theta_n^N), \delta_{\bar{\theta}_\star})$, we provide a result adapted from Altschuler and Chewi [2023, Lemma A.8] to our non-differentiable setting that will be useful to bound the first term of (28).

Lemma A.14. *Suppose that the distribution $\pi \propto \exp(-f)$ on $\mathbb{R}^d$ is α-strongly log-concave, almost everywhere differentiable and that $x^\star$ is the minimiser of f. Then,*

$$\mathbb{E}_{X \sim \pi}[\|X - x^\star\|^2] \le d/\alpha.$$

Proof. Let $\Omega \subset \mathbb{R}^d$ denote the set of differentiable points of f, note that when f is convex and differentiable at $x \in \Omega$, then $\partial f(x) = \{\nabla f(x)\}$, that is, its gradient is its only subgradient. Recall also that f is strongly convex, so for every $x, y \in \mathbb{R}^d$ we have that

$$\langle \partial f(x), x - y \rangle \ge \alpha \|x - y\|^2.$$

Integration by parts shows that for any smooth function $\phi : \mathbb{R}^d \to \mathbb{R}$ of controlled growth, it holds that

$$0 = \int_\Omega (\Delta\phi - \langle \nabla f, \nabla \phi \rangle)\, \mathrm{d}\pi = \mathbb{E}_{X \sim \pi}[\Delta\phi - \langle \nabla f, \nabla \phi \rangle]. \tag{38}$$

Applying (38) to the function $\phi(x) := \|x - x^\star\|^2/2$, for which $\nabla\phi(x) = x - x^\star$ and $\Delta\phi = d$, together with the strong convexity of f, the result follows. □

To conclude we present the following theorem that provides a convergence bound for PIPGLA in terms of N, γ, n, λ and the convexity properties of U.

Theorem A.15. *[Theorem 4.2 restated] Let **A1**, **A2**, **A4** and **C1** hold. Then for $\gamma \leq 1/L_{g_1}$ and $\gamma \leq \lambda \leq \gamma/(1-\mu\gamma)$, PIPGLA satisfies*

$$W_2(\mathcal{L}(\theta_n^N), \delta_{\bar{\theta}_\star}) \leq \sqrt{\frac{d_\theta}{N\mu}} + \frac{\lambda^{n/2}(1-\gamma\mu)^{n/2}}{\gamma^{n/2}} W_2(\mathcal{L}(Z_0^N), \pi^N) + \left(\frac{\lambda(2\gamma L_{g_1}(d_\theta + Nd_x) + \lambda NC)}{N\,(1-\lambda(1-\mu\gamma)/\gamma)}\right)^{1/2}$$

*for all $n \in \mathbb{N}$, with Z_0^N given in **A2** and $C > 0$ given in **C1** and independent of $t, n, N, \gamma, d_\theta, d_x$.*

Proof. Using a form of the Prékopa-Leindler inequality for strong convexity [Saumard and Wellner, 2014, Theorem 3.8], π_Θ is μ-strongly log-concave. Therefore, π_Θ^N is $N\mu$-strongly log-concave and satisfies all the assumptions of Lemma A.14. So, we have that

$$W_2(\delta_{\bar{\theta}_*}, \pi_\Theta^N)^2 \leq \frac{d_\theta}{N\mu}.$$

On the other hand, note that $\pi^N(z) \propto \exp(-N(G_1(z) + G_2(z))) = \exp(-\sum_i U(z_\theta, \sqrt{N} z_i))$. By Corollary A.13 it follows that

$$W_2(\mathcal{L}(\theta_n^N), \pi_\Theta^N) \leq W_2(\mathcal{L}(Z_n), \pi^N) \leq \sqrt{\frac{\lambda^n(1-\gamma\mu)^n}{\gamma^n} W_2^2(\mathcal{L}(Z_0^N), \pi^N) + \frac{\lambda(2\gamma L_{g_1}(d_\theta + Nd_x) + \lambda NC)}{N\,(1-\lambda(1-\mu\gamma)/\gamma)}}.$$

Using that $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$, we have

$$W_2(\mathcal{L}(\theta_n^N), \pi_\Theta^N) \leq \frac{\lambda^{n/2}(1-\gamma\mu)^{n/2}}{\gamma^{n/2}} W_2(\mathcal{L}(Z_0^N), \pi^N) + \left(\frac{\lambda(2\gamma L_{g_1}(d_\theta + Nd_x) + \lambda NC)}{N\,(1-\lambda(1-\mu\gamma)/\gamma)}\right)^{1/2}.$$

The proof then follows from (28) and the above. □

B THEORETICAL ANALYSIS OF PROXIMAL PARTICLE GRADIENT DESCENT

B.1 BACKGROUND ON PARTICLE GRADIENT DESCENT

The PGD algorithm [Kuntz et al., 2023] relies on the perspective that the MMLE problem can be solved by minimising the free energy

$$F(\theta, q) = \int \log\big(q(x)\big) q(x) \mathrm{d}x + \int U(\theta, x) q(x) \mathrm{d}x \tag{39}$$

for all $(\theta, q) \in \Theta \times \mathcal{P}(\mathbb{R}^{d_x})$, where Θ denotes the parameter space and $U(\theta, x) \coloneqq -\log p_\theta(x, y)$. Kuntz et al. [2023] propose a discretisation of a gradient flow associated with (39), where they endow Θ with the Euclidean geometry and $\mathcal{P}(\mathbb{R}^{d_x})$ with the 2-Wasserstein one to take gradients. This leads to the Euclidean-Wasserstein gradient flow of F

$$\begin{aligned} \dot{\boldsymbol{\theta}}_t &= -\nabla_\theta F(\boldsymbol{\theta}_t, q_t) = -\int \nabla_\theta U(\boldsymbol{\theta}_t, x) q_t(x) \mathrm{d}x, \\ \dot{q}_t &= -\nabla_q F(\boldsymbol{\theta}_t, q_t) = \nabla_x \cdot \Big[q_t \nabla_x \log\Big(\frac{q_t}{p_{\boldsymbol{\theta}_t}(\cdot, y)}\Big)\Big]. \end{aligned} \tag{40}$$

Kuntz et al. [2023] prove that the gradient $\nabla F(\theta, q)$ vanishes if and only if θ is a stationary point of $p_\theta(y)$ and q is its corresponding posterior. Based on the observation that (40) is a Fokker-Planck equation satisfied by the law of a McKean-Vlasov SDE, and using a finite number of particles $(X_t^{i,N})_{i=1}^N$ to estimate q_t, they obtain the following approximation, for $t \geq 0$,

$$\begin{aligned} \mathrm{d}\boldsymbol{\theta}_t^N &= -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N}) \mathrm{d}t, \\ \mathrm{d}\mathbf{X}_t^{i,N} &= -\nabla_x U(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N}) \mathrm{d}t + \sqrt{2} \mathrm{d}\mathbf{B}_t^{i,N}, \qquad i = 1, \ldots, N, \end{aligned} \tag{41}$$

Algorithm 1 Moreau-Yosida Particle Gradient Descent (MYPGD)

Require: $N, K, \lambda, \gamma, \pi_{\text{init}} \in \mathcal{P}(\mathbb{R}^{d_\theta}) \times \mathcal{P}((\mathbb{R}^{d_x})^N)$
Draw $(\theta_0, \{X_0^{i,N}\}_{i=1}^N)$ from π_{init}
for $n = 0 : K$ **do**

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

end for
return θ_{K+1}^N

where $(\mathbf{B}_t^{i,N})_{t\geq 0}$ for $i = 0, \dots, N$ are d_x-dimensional Brownian motions. Using a simple Euler–Maruyama discretisation with step size $\gamma > 0$ of (41) one obtains the particle gradient descent (PGD) algorithm [Kuntz et al., 2023]

$$\theta_{n+1} = \theta_n - \frac{\gamma}{N}\sum_{j=1}^N \nabla_\theta U(\theta_n, X_n^{j,N}),$$
$$X_{n+1}^{i,N} = X_n^{i,N} - \gamma\nabla_x U(\theta_n, X_n^{i,N}) + \sqrt{2\gamma}\xi_{n+1}^{i,N}, \qquad i = 1, \dots, N,$$

where (ξ_n) for $n \geq 0$ are d_x-dimensional i.i.d. standard Gaussians.

B.2 PROXIMAL PARTICLE GRADIENT DESCENT

Similar to the approach we have taken in the main text, we can also provide a proximal version of the PGD algorithm. As mentioned in the main text, if we remove the noise term in the dynamics of θ, we obtain

$$\mathrm{d}\boldsymbol{\theta}_t^N = -\frac{1}{N}\sum_{i=1}^N \nabla_\theta U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t, \tag{42}$$
$$\mathrm{d}\mathbf{X}_t^{i,N} = -\nabla_x U^\lambda(\boldsymbol{\theta}_t^N, \mathbf{X}_t^{i,N})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{i,N}. \tag{43}$$

We can then provide an algorithm which is a discretisation of (42)-(43), termed Moreau-Yosida Particle Gradient Descent (MYPGD), analogous to MYIPLA. The algorithm is given in Algorithm 1.

We extend the results of Caprio et al. [2024] to provide a nonasymptotic bound for MYPGD. To do so, we consider the following metric on $\mathbb{R}^{d_\theta} \times \mathcal{P}_2(\mathbb{R}^{d_x})$

$$\mathbf{d}((\theta, q), (\theta', q')) = \sqrt{\|\theta - \theta'\|^2 + W_2^2(q, q')}.$$

Under similar assumptions to those used in Theorem 4.1 we obtain the following result.

Theorem B.1 (MYPGD)**.** *Let **A1–A5** hold. If $X_0^1, \dots, X_0^N$ are drawn independently from a distribution q_0 in $\mathcal{P}_2(\mathbb{R}^{d_x})$ and $\lambda > 0$, $\gamma \leq 1/(L_{g_1} + \lambda^{-1} + \mu)$, then*

$$\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} \leq \frac{\lambda}{\mu}\Big(\frac{\|g_2\|_{\text{Lip}}^2}{2}A + B\Big) + \frac{(L_{g_1} + \lambda^{-1})\sqrt{2}}{\mu\sqrt{N}}\sqrt{B_0 + \frac{2d_x}{\mu}}$$
$$+ \mathbf{d}((\theta_0, q_0), (\bar{\theta}_{\star,\lambda}, \pi_{\star,\lambda}))e^{-\mu n\gamma} + A_{0,\gamma,\lambda}\gamma^{1/2} + \mathcal{O}(\lambda^2)$$

for all $n \in \mathbb{N}$; where $B_0 = \|\theta_0\|^2 + \sup_{i\in 1,\dots,N}\mathbb{E}[\|X_0^{i,N}\|^2] < \infty$ and

$$A_{0,\gamma,\lambda} = \sqrt{\frac{4\gamma + 4/a}{a}220(L_{g_1} + \lambda^{-1})^2\Big(\gamma(L_{g_1} + \lambda^{-1})^2\Big[B_0 + \frac{2d_x}{\mu}\Big] + d_x\Big)}, \qquad a = \frac{2(L_{g_1} + \lambda^{-1})\mu}{L_{g_1} + \lambda^{-1} + \mu}.$$

Proof. Let us denote by $(\theta_n^N, Q_n^{N,\gamma})$ the MYPGD output after n iterations using a discretisation step γ and by $Q_{\star,\lambda}^N$ the empirical distribution of N i.i.d. particles drawn from $\pi_{\bar{\theta}_{\star,\lambda}}$. Using the triangular inequality, we have

$$\mathbb{E}[\|\theta_n^N - \bar{\theta}_\star\|^2]^{1/2} \leq \|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| + \mathbb{E}[\|\theta_n^N - \bar{\theta}_{\star,\lambda}\|^2]^{1/2} \leq \|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\| + \mathbf{d}((\theta_n^N, Q_n^{N,\gamma}), (\bar{\theta}_{\star,\lambda}, Q_{\star,\lambda}^N)).$$

The term $\|\bar{\theta}_\star - \bar{\theta}_{\star,\lambda}\|$ can be upper bounded by $\frac{\lambda}{\mu}\left(\frac{\|g_2\|_{\mathrm{Lip}}^2}{2}A + B\right) + \mathcal{O}(\lambda^2)$ using Proposition A.1, while a bound for the second term $\mathbf{d}((\theta_n^N, Q_n^{N,\gamma}), (\bar{\theta}_{\star,\lambda}, Q_{\star,\lambda}^N))$ is derived in Caprio et al. [2024, Theorem 7], which gives the desired result. □

Selecting $\gamma = \lambda$ and $g_1 = 0$ in MYPGD we obtain an extension of PGD corresponding to the PIPULA algorithm introduced in Section 3.1.1, that we termed Proximal PGD (PPGD). Obtaining a rigorous bound like that in Theorem B.1 for this algorithm is more challenging due to the presence of γ both as time discretisation parameter and in the Lipschitz constant of ∇U^γ. In particular, while under **A1**–**A3**, **A5** and **B1** Caprio et al. [2024, Lemma 10 and 11] hold with $\lambda = \gamma$, establishing a result controlling the time discretisation error like that in Caprio et al. [2024, Lemma 12] is not straightforward.

C CONVERGENCE TO WASSERSTEIN GRADIENT FLOW

We now show that the continuous time interacting particle system introduced in (7)–(8) converges in the large N limit (i.e. $N \to \infty$) to a McKean–Vlasov SDE with a solution whose law satisfies the Euclidean-Wasserstein gradient flow

$$\begin{aligned}
\dot{\boldsymbol{\theta}}_{\lambda,t} &= -\nabla_\theta F(\boldsymbol{\theta}_{\lambda,t}, q_{\lambda,t}) = -\int \nabla_\theta U^\lambda(\boldsymbol{\theta}_{\lambda,t}, x) q_{\lambda,t}(x) \mathrm{d}x, \\
\dot{q}_{\lambda,t} &= -\nabla_q F(\boldsymbol{\theta}_{\lambda,t}, q_{\lambda,t}) = \nabla_x \cdot \left[q_{\lambda,t} \nabla_x \log\left(\frac{q_{\lambda,t}}{p^\lambda_{\boldsymbol{\theta}_{\lambda,t}}(\cdot, y)} \right) \right],
\end{aligned}$$

where $p^\lambda_{\boldsymbol{\theta}_{\lambda,t}}$ denotes the Moreau-Yosida envelope of $p_{\boldsymbol{\theta}_{\lambda,t}}$. This result is classical in the study of McKean–Vlasov SDEs, where is referred to as *propagation of chaos* (e.g. Sznitman [1991, Theorem 1.4]).

We start by proving the following auxiliary result. Let us denote, for any $\theta \in \mathbb{R}^{d_\theta}$ and $\nu \in \mathcal{P}(\mathbb{R}^{d_x})$, $g(\theta, \nu) := \int_{\mathbb{R}^{d_x}} \nabla_\theta U^\lambda(\theta, x')\nu(x')\mathrm{d}x'$.

Lemma C.1. *The function $g : \mathbb{R}^{d_\theta} \times \mathcal{P}(\mathbb{R}^{d_x}) \to \mathbb{R}^{d_\theta}$ is Lipschitz continuous in both arguments, i.e.,*

$$\|g(\theta_1, \nu_1) - g(\theta_2, \nu_2)\| \leq \lambda^{-1} \left(\|\theta_1 - \theta_2\| + W_1(\nu_1, \nu_2) \right).$$

Proof. Rockafellar and Wets [2009, Proposition 12.19] shows that ∇U^λ is Lipschitz continuous with Lipschitz constant λ^{-1}. Then the result follows from Akyildiz et al. [2025, Lemma 5]. □

We can now show convergence of (7)–(8) to the following McKean–Vlasov SDE

$$\begin{aligned}
\mathrm{d}\boldsymbol{\theta}_{\lambda,t} &= -\left[\int \nabla_\theta U^\lambda(\boldsymbol{\theta}_{\lambda,t}, x) q_{\lambda,t}(x) \mathrm{d}x \right] \mathrm{d}t \\
\mathrm{d}\mathbf{X}_{\lambda,t} &= -\nabla_x U^\lambda(\boldsymbol{\theta}_{\lambda,t}, \mathbf{X}_{\lambda,t}) \mathrm{d}t + \sqrt{2} \mathrm{d}\mathbf{B}_t.
\end{aligned} \tag{44}$$

Proposition C.2 (Propagation of chaos)**.** *For any (exchangeable) initial condition $(\theta_0^N, X_0^{1:N})$ such that $(\theta_0^N, X_0^{j,N}) = (\theta_0, X_0)$ for $j = 1, \ldots, N$ with $\mathbb{E}\left[|\theta_0|^2 + |X_0|^2\right] < \infty$, we have for any $T \geq 0$*

$$\mathbb{E}\left[\sup_{t \in [0,T]} \left(\|\boldsymbol{\theta}_{\lambda,t} - \boldsymbol{\theta}_t^N\| + \|\mathbf{X}_{\lambda,t} - \mathbf{X}_t^{j,N}\| \right) \right] \leq \frac{\sqrt{2}(\sqrt{C_T}\lambda^{-1} + \sqrt{T}) e^{2T\lambda^{-1}}}{N^{1/2}} \tag{45}$$

where $C_T := \sup_{t \leq T} \mathbb{E}\left[|\boldsymbol{\theta}_t^N|^2 + |\mathbf{X}_t^{j,N}|^2 \right] < \infty$, for any $j = 1, \ldots, N$.

Proof. The proof exploits the Lipschitz continuity of ∇U^λ and of g established in Lemma C.1. The argument is classical and omitted, see Akyildiz et al. [2025, Proposition 8] for the proof in a similar context. □

We can further show that (44) converges to the following MKVSDE

$$\begin{aligned} \mathrm{d}\boldsymbol{\theta}_t &= -\left[\int \nabla_\theta U(\boldsymbol{\theta}_t, x) q_t(x) \mathrm{d}x\right] \mathrm{d}t \\ \mathrm{d}\mathbf{X}_t &= -\nabla_x U(\boldsymbol{\theta}_t, \mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \end{aligned} \tag{46}$$

associated with the gradient flow (40), when $\lambda \to 0$.

Proposition C.3. *Assume that U is gradient Lipschitz with constant $\|\nabla U\|_{\mathrm{Lip}}$. For any initial condition (θ_0, X_0) such that $\mathbb{E}\left[|\theta_0|^2 + |X_0|^2\right] < \infty$, we have for any $T \geq 0$*

$$\mathbb{E}\left[\sup_{t\in[0,T]} \left(\|\boldsymbol{\theta}_{\lambda,t} - \boldsymbol{\theta}_t\|^2 + \|\mathbf{X}_{\lambda,t} - \mathbf{X}_t\|^2\right)\right] \leq \left(\lambda^2 \|\nabla U\|_{\mathrm{Lip}}^4 C_T + \|\nabla U\|_{\mathrm{Lip}}^2 \mathcal{O}(\lambda^4)\right) T \exp(2\|\nabla U\|_{\mathrm{Lip}}^2 T),$$

where $C_T := \sup_{t\leq T} \mathbb{E}\left[|\boldsymbol{\theta}_{\lambda,t}|^2 + |\mathbf{X}_{\lambda,t}|^2\right] < \infty$. It follows that, as $\lambda \to 0$, (44) converges to (46) in $\mathbb{L}^2$.

Proof. For any $t \geq 0$, we have

$$\begin{aligned} \boldsymbol{\theta}_t &= \theta_0 + \int_0^t \left[-\int \nabla_\theta U(\boldsymbol{\theta}_s, x) q_s(x) \mathrm{d}x\right] \mathrm{d}s, \\ \mathbf{X}_t &= X_0 - \int_0^t \nabla_x U(\boldsymbol{\theta}_s, \mathbf{X}_s)\mathrm{d}s + \sqrt{2}\boldsymbol{B}_t, \end{aligned}$$

and equivalently for $\boldsymbol{\theta}_{\lambda,t}, X_{\lambda,t}$. We first observe that

$$\begin{aligned} \|\boldsymbol{\theta}_{\lambda,t} - \boldsymbol{\theta}_t\|^2 &= \left\| \int_0^t \left[\int \nabla_\theta U^\lambda(\boldsymbol{\theta}_{\lambda,s}, x) q_{\lambda,s}(x)\mathrm{d}x - \int \nabla_\theta U(\boldsymbol{\theta}_s, x) q_s(x)\mathrm{d}x\right] \mathrm{d}s \right\|^2 \\ &= \left\| \int_0^t \mathbb{E}\left[\nabla_\theta U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s}) - \nabla_\theta U(\boldsymbol{\theta}_s, \mathbf{X}_s)\right] \mathrm{d}s \right\|^2 \\ &\leq \mathbb{E}\left[\left\| \int_0^t [\nabla_\theta U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s}) - \nabla_\theta U(\boldsymbol{\theta}_s, \mathbf{X}_s)]\mathrm{d}s \right\|^2 \right], \end{aligned}$$

and

$$\mathbb{E}\left[\|\mathbf{X}_{\lambda,t} - \mathbf{X}_t\|^2\right] = \mathbb{E}\left[\left\| \int_0^t [\nabla_x U(\boldsymbol{\theta}_s, \mathbf{X}_s) - \nabla_x U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})]\mathrm{d}s \right\|^2 \right].$$

Combining the above we obtain

$$\begin{aligned} \mathbb{E}\Big[\sup_{s\in[0,t]} \|\boldsymbol{\theta}_{\lambda,s} - \boldsymbol{\theta}_s\|^2 + \mathbb{E}[\|\mathbf{X}_{\lambda,s} - \mathbf{X}_s\|^2] \Big] &\leq \mathbb{E}\left[\int_0^t \|\nabla U(\boldsymbol{\theta}_s, \mathbf{X}_s) - \nabla U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2 \mathrm{d}s \right] \\ &\leq 2\mathbb{E}\left[\int_0^t \|\nabla U(\boldsymbol{\theta}_s, \mathbf{X}_s) - \nabla U(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2 \mathrm{d}s \right] \\ &+ 2\mathbb{E}\left[\int_0^t \|\nabla U(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s}) - \nabla U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2 \mathrm{d}s \right] \\ &\leq 2\|\nabla U\|_{\mathrm{Lip}}^2 \int_0^t \left(\|\boldsymbol{\theta}_{\lambda,s} - \boldsymbol{\theta}_s\|^2 + \mathbb{E}[\|\mathbf{X}_{\lambda,s} - \mathbf{X}_s\|^2]\right) \mathrm{d}s \\ &+ 2\mathbb{E}\left[\int_0^t \|\nabla U(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s}) - \nabla U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2 \mathrm{d}s \right]. \end{aligned}$$

In the case in which ∇U is Lipschitz continuous, we further have that $\nabla U^\lambda(v) = \nabla U(\mathrm{prox}_U^\lambda(v))$ for $v = (\theta, x)$ [Pereyra, 2016, Section 2], and we have

$$\|\nabla U(v) - \nabla U^\lambda(v)\| = \|\nabla U(v) - \nabla U(\mathrm{prox}_U^\lambda(v))\| \leq \|\nabla U\|_{\mathrm{Lip}} \|v - \mathrm{prox}_U^\lambda(v)\|.$$

Recalling that, since U is gradient Lipschitz, $\mathrm{prox}_U^\lambda(v) = v - \lambda\nabla U(v) + \mathcal{O}(\lambda^2)$ [Parikh and Boyd, 2014, Section 3.3], we further have that

$$\begin{aligned}\|\nabla U(v) - \nabla U^\lambda(v)\| &\leq \|\nabla U\|_{\text{Lip}}(\lambda\|\nabla U(v)\| + \mathcal{O}(\lambda^2)) \\ &\leq \|\nabla U\|_{\text{Lip}}(\lambda\|\nabla U\|_{\text{Lip}}(1 + \|v\|) + \mathcal{O}(\lambda^2)),\end{aligned} \tag{47}$$

and we can bound

$$\begin{aligned}&\mathbb{E}\left[\int_0^t \|\nabla U(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s}) - \nabla U^\lambda(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2 \mathrm{d}s\right] \\ &\leq \lambda^2\|\nabla U\|_{\text{Lip}}^4 \int_0^t \mathbb{E}\left[1 + \|(\boldsymbol{\theta}_{\lambda,s}, \mathbf{X}_{\lambda,s})\|^2\right] \mathrm{d}s + \|\nabla U\|_{\text{Lip}}^2 \mathcal{O}(\lambda^4)t \\ &\leq \lambda^2\|\nabla U\|_{\text{Lip}}^4 C_T t + \|\nabla U\|_{\text{Lip}}^2 \mathcal{O}(\lambda^4)t\end{aligned}$$

with C_T given in the statement of the result.

Let us denote by $h(t) = \sup_{s\in[0,t]} \|\boldsymbol{\theta}_{\lambda,s} - \boldsymbol{\theta}_s\|^2 + \mathbb{E}[\|\mathbf{X}_{\lambda,s} - \mathbf{X}_s\|^2]$. Then, using the bounds above we have that

$$h(t) \leq 2\|\nabla U\|_{\text{Lip}}^2 \int_0^t h(s)\mathrm{d}s + \left(\lambda^2\|\nabla U\|_{\text{Lip}}^4 C_T + \|\nabla U\|_{\text{Lip}}^2 \mathcal{O}(\lambda^4)\right) t.$$

Using Gronwall's inequality we obtain

$$h(t) \leq \left(\lambda^2\|\nabla U\|_{\text{Lip}}^4 C_T + \|\nabla U\|_{\text{Lip}}^2 \mathcal{O}(\lambda^4)\right) t \exp(2\|\nabla U\|_{\text{Lip}}^2 t),$$

from which the result follows. □

D ALGORITHM COMPARISON

We provide further details on computing the complexity estimates of Section 4.5. For convenience, we summarise the complexity estimates for λ, the number of particles N, γ, and the number of steps n to achieve an error $\mathbb{E}\left[\|\theta_n^N - \bar{\theta}^\star\|^2\right]^{1/2} = \mathcal{O}(\varepsilon)$ from Table 1. The values are provided in terms of the key parameters d_θ, d_x and $\delta > 0$ is any small positive constant.

	λ	N	γ	n
MYIPLA	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(d_\theta\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x\varepsilon^{-2-\delta})$
PIPGLA	$\mathcal{O}(\varepsilon^2)$	$\mathcal{O}(d_\theta\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(\log\varepsilon^2/\log d_x)$
MYPGD	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(d_x\varepsilon^{-2})$	$\mathcal{O}(d_x^{-1}\varepsilon^2)$	$\mathcal{O}(d_x\varepsilon^{-2-\delta})$

The bound for MYIPLA follows from first choosing λ so that the first term in Theorem 4.1 is $\mathcal{O}(\varepsilon)$, then choosing N so that the second term is $\mathcal{O}(\varepsilon)$ and γ sufficiently small to counteract the dependence on d_x in the fourth term. Finally, since for every $p \in \mathbb{N}$ one has $e^x \geq x^p/p!$ for $x > 0$, for every $\delta > 0$ (by choosing $p \in \mathbb{N}$ large enough) one has $e^{-\varepsilon^\delta} \leq C\varepsilon$. Therefore, as long as n is chosen sufficiently large that $\mu n\gamma = \mathcal{O}(\varepsilon^{-\delta})$, the exponential decay is strong enough so that the middle term is of order $\mathcal{O}(\varepsilon)$. A similar approach based on the bound in Theorem B.1 provides the bounds for MYPGD.

On the other hand, the bound for PIPGLA follows from first choosing N so that the first term in Theorem 4.2 is $\mathcal{O}(\varepsilon)$, then λ and γ to counteract the dependence of d_x on the third term. Finally, considering the values of λ and γ, we select n to ensure that the second term is $\mathcal{O}(\varepsilon)$.

On the other hand, we also compared the algorithms in terms of their computational requirements. We account for the computational cost of running each algorithm for n iterations with N particles and time discretisation step γ, while guaranteeing an $\mathcal{O}(\varepsilon)$ error, in terms of component-wise evaluations of ∇g_1 and $\mathrm{prox}_{g_2}^\lambda$, and number of independent standard 1-dimensional Gaussian samples.

For every step of MYIPLA, PIPGLA and MYPGD one requires $N(d_\theta + d_x)$ evaluations of ∇g_1 component-wise and $N(d_\theta + d_x)$ evaluations of $\mathrm{prox}_{g_2}^\lambda$ component-wise. In the case of MYIPLA and PIPGLA, we need $d_\theta + Nd_x$ independent standard 1-dimensional Gaussians for each iteration; since MYPGD does not have a noise in the θ-component this reduces to Nd_x.

	Evaluations of ∇g_1	Evaluations of $\text{prox}_{g_2}^{\lambda}$	Indep. 1d Gaussians
MYIPLA	$\mathcal{O}(d_\theta d_x(d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_\theta d_x(d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_\theta d_x^2\varepsilon^{-4-\delta})$
PIPGLA	$\mathcal{O}(d_\theta(d_\theta + d_x)\varepsilon^{-2}\frac{\log\varepsilon^2}{\log d_x})$	$\mathcal{O}(d_\theta(d_\theta + d_x)\varepsilon^{-2}\frac{\log\varepsilon^2}{\log d_x})$	$\mathcal{O}(d_\theta d_x\varepsilon^{-2}\frac{\log\varepsilon^2}{\log d_x})$
MYPGD	$\mathcal{O}(d_x^2(d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_x^2(d_\theta + d_x)\varepsilon^{-4-\delta})$	$\mathcal{O}(d_x^3\varepsilon^{-4-\delta})$

Finally, Table 4 summarises the key differences and advantages of each algorithm. We recall that L_{g_1} and μ denote the Lipschitz continuity and strong-convexity parameters of g_1, introduced in assumptions **A1** and **A4**, respectively.

Table 4: Comparison of convergence assumptions, parameter constraints, and advantages of each algorithm.

	Assumptions for convergence	**Constraints on λ**	**Constraints on γ**	**Advantages**
MYIPLA	**A1**–**A5**	$\lambda \geq 0$	$\gamma < \min\left\{(L_{g_1} + \lambda^{-1})^{-1},\, 2\mu^{-1}\right\}$	Noise in the θ-dynamics helps escape local minima.
PIPGLA	**A1**, **A2**, **A4** and **C1**	$\gamma \leq \lambda \leq \gamma/(1-\mu\gamma)$	$\gamma \leq 1/L_{g_1}$	Ensures θ estimates remain within the support of the distribution.
MYPGD	**A1**–**A5**	$\lambda \geq 0$	$\gamma < (L_{g_1} + \lambda^{-1} + \mu)^{-1}$	Produces lower-variance estimates for MMLE in the strongly convex setting.

E NUMERICAL EXPERIMENTS

E.1 DERIVATION OF THE PROXIMAL OPERATORS

E.1.1 Laplace Prior with Unknown Mean θ

We recall that using a Laplace prior $g_2(\theta, x) = \sum_{i=1}^{d_x} |x_i - \theta|$.

$$\text{prox}_{g_2}^{\lambda}(\theta, x) = \underset{(u_0,u)}{\arg\min}\, h(u_0, u) = \underset{(u_0,u)}{\arg\min}\{g_2(u_0, u) + \|(u_0, u) - (\theta, x)\|^2/(2\lambda)\}.$$

The first order optimality condition is given by

$$0 \in \partial g_2(u_0, u) + \nabla\big(\|(u_0, u) - (\theta, x)\|^2/(2\lambda)\big).$$

We recall that $\phi \in \mathbb{R}^d$ is a subdifferential of the ℓ^1-norm at $x \in \mathbb{R}^d$ if and only if $\phi_i(x) = \text{sign}(x_i)$ if $x_i \neq 0$ and $|\phi_i(x)| \leq 1$ otherwise [Parikh and Boyd, 2014].

Let us define the set $D = \{i \in \{1, \ldots, d_x\} | u_i - u_0 = 0\}$. Then, the first order optimality condition becomes

$$0 \in \left\{ -\sum_{i \notin D} t_i - \sum_{i \notin D} \text{sign}(u_i - u_0) + (u_0 - \theta)/\lambda \mid |t_i| \leq 1 \right\}$$
$$\begin{cases} 0 \in \left\{ t_i + \frac{u_i - x_i}{\lambda} \mid |t_i| \leq 1 \right\} & \text{if } i \in D \\ 0 = \text{sign}(u_i - u_0) + (u_i - x_i)/\lambda & \text{if } i \notin D \end{cases}.$$

Reordering terms, we get

$$u_0 \in \left\{ \theta + \lambda\Big(\sum_{i \notin D} t_i - \sum_{i \notin D} \text{sign}(u_i - u_0) \Big) \mid |t_i| \leq 1 \right\}, \tag{48}$$

$$\begin{cases} u_i \in \{x_i - \lambda t_i \mid |t_i| \leq 1\} & \text{if } i \in D, \\ u_i = x_i - \lambda\, \text{sign}(u_i - u_0) & \text{if } i \notin D. \end{cases} \tag{49}$$

Assuming that $D = \emptyset$, the previous system of equations can be solved iteratively using a fixed point algorithm.

Alternatively, for a lower computational cost we can obtain an approximate solution by setting $u_0 = \theta$ in (49)

$$\begin{cases} u_i \in \{x_i - \lambda t_i \mid |t_i| \leq 1\} & \text{if } i \in D, \\ u_i = x_i - \lambda \operatorname{sign}(u_i - \theta) & \text{if } i \notin D. \end{cases}$$

which is solved applying the soft-thresholding operator

$$u_i = \theta + [x_i - \theta - \lambda \operatorname{sign}(x_i - \theta)]\mathbb{1}\{|x_i - \theta| \geq \lambda\}.$$

Using these u_i's, taking $u_0 = \theta$ in the right-hand side of (48) and assuming $D = \emptyset$, we obtain

$$u_0 = \theta + \lambda \sum_{i=1}^{d_x} \operatorname{sign}(u_i - \theta).$$

E.1.2 Laplace Prior with Unknown Scale $e^{2\theta}$

For the Bayesian neural network experiment we consider a Laplace prior with zero mean and unknown scale parameterised by $e^{2\theta}$ (which ensures that the scale is positive), we have $g_2(\theta, x) = d_x\alpha + \sum_i |x_i|e^{-2\alpha}$. Its proximal operator is given by

$$\operatorname{prox}_{g_2}^{\lambda}(\theta, x) = \underset{(u_0,u)}{\arg\min}\, h(u_0, u), \quad h(u_0, u) = u_0 d_x + \sum_i |u_i|e^{-2u_0} + \|(u_0, u) - (\theta, x)\|^2/(2\lambda).$$

The optimality condition is given by

$$0 \in \partial\big(u_0 d_x + \sum_i |u_i|e^{-2u_0}\big) + \nabla\big(\|(u_0, u) - (\theta, x)\|^2/(2\lambda)\big),$$

which provides the following system of equations

$$0 = d_x - 2e^{-2u_0} \sum_{i=1}^{d_x} |u_i| + \frac{1}{\lambda}(u_0 - \theta),$$

$$\begin{cases} 0 \in \big\{e^{-2u_0} t_i + (u_i - x_i)/\lambda \mid |t_i| \leq 1\big\} & \text{if } u_i = 0, \\ 0 = e^{-2u_0} \operatorname{sign}(u_i) + (u_i - x_i)/\lambda & \text{if } u_i \neq 0. \end{cases}$$

Reordering terms, we get

$$u_0 = \theta - \lambda d_x + 2\lambda e^{-2u_0} \sum_i |u_i| \tag{50}$$

$$\begin{cases} u_i \in \{x_i - \lambda e^{-2u_0} t_i \mid |t_i| \leq 1\} & \text{if } u_i = 0, \\ u_i = x_i - \lambda e^{-2u_0} \operatorname{sign}(u_i) & \text{if } u_i \neq 0. \end{cases} \tag{51}$$

This system of equations can be solved using an iterative solver, however this will incur in a high computational cost. Therefore, we opt for the following approximation of (51), where we set $u_0 = \theta$,

$$\begin{cases} u_i \in \{x_i - \lambda e^{-2\theta} t_i \mid |t_i| \leq 1\} & \text{if } u_i = 0, \\ u_i = x_i - \lambda e^{-2\theta} \operatorname{sign}(u_i) & \text{if } u_i \neq 0. \end{cases} \tag{52}$$

The solution of (52) is

$$u_i \approx [x_i - \lambda e^{-2\theta} \operatorname{sign}(x_i)]\mathbb{1}\{|x_i| \geq \lambda e^{-2\theta}\}.$$

Using these u_i's together with the Lambert W function, the solution of (50) is given by

$$u_0 \approx \theta - \lambda d_x + \frac{1}{2} W\left(4\lambda e^{-2\theta} \sum_i |u_i|\right).$$

E.1.3 Uniform Prior

We recall that using a uniform prior

$$g_2(\theta, x) = d_x \log(2\theta) + \sum_{i=1}^{d_x} \imath_{[-\theta,\theta]}(x_i),$$

where $\imath_{\mathcal{K}}$ is the convex indicator of $\mathcal{K}$ defined by $\imath_{\mathcal{K}}(x) = 0$ if $x \in \mathcal{K}$ and $\imath_{\mathcal{K}}(x) = \infty$ otherwise. In this case, the proximal operator satisfies

$$\begin{aligned}\operatorname{prox}^{\lambda}_{g_2}(\theta, x) &= \arg\min_{(u_0,u)} \{g_2(u_0, u) + \|(u_0, u) - (\theta, x)\|^2/(2\lambda)\} \\ &= \arg\min_{\substack{(u_0,u) \\ |u_i| \le u_0}} \{d_x \log(2u_0) + \|(u_0, u) - (\theta, x)\|^2/(2\lambda)\}.\end{aligned}$$

We can obtain an approximate solution by deriving the first order conditions for u_i with $i = 0, 1, \ldots, d_x$ and combining them with the constraint $|u_i| \le u_0$:

$$\begin{aligned}u_0 &= \begin{cases} \frac{\theta + \sqrt{\theta^2 - 4\lambda d_x}}{2} & \text{if } \theta^2 \ge 4\lambda d_x, \\ \max_i |x_i| & \text{otherwise,} \end{cases} \\ u_i &= \operatorname{sign}(x_i) \cdot \min\{|x_i|, |u_0|\}.\end{aligned}$$

E.1.4 Approximation for PIPULA and PPGD

In PIPULA and PPGD, we need to compute the proximal operator of $U = g_1 + g_2$ which is usually not available in closed form. Since γ is normally set to a small enough value, we follow Pereyra [2016] and approximate the proximity map of U as

$$\begin{aligned}\operatorname{prox}^{\gamma}_{U}(v) &= \arg\min_{v'} \{g_1(v') + g_2(v') + \|v' - v\|^2/(2\gamma)\} \\ &\approx \arg\min_{v'} \{g_1(v) + (v' - v)^\intercal \nabla g_1(v) + g_2(v') + \|v' - v\|^2/(2\gamma)\} \\ &\approx \arg\min_{v'} \{g_2(v') + \|v' - v + 2\gamma \nabla g_1^\intercal(v)\|^2/(2\gamma)\} \\ &\approx \operatorname{prox}^{\gamma}_{g_2}(v + 2\gamma \nabla g_1^\intercal(v)),\end{aligned}$$

where $v = (\theta, x), v' = (\theta', x')$.

E.2 BAYESIAN LOGISTIC REGRESSION

In the case of the Laplace prior, the negative log joint likelihood is given by

$$-\log p_\theta(x, y) = \underbrace{\sum_{i=1}^{d_x} |x_i - \theta|}_{g_2(\theta,x)} + \underbrace{d_x \log 2 - \log p(y|x)}_{g_1(\theta,x)};$$

and for the uniform prior, we obtain

$$-\log p_\theta(x, y) = \underbrace{d_x \log(2\theta) + \sum_{i=1}^{d_x} \imath_{[-\theta,\theta]}(x_i)}_{g_2(\theta,x)} \underbrace{- \log p(y|x)}_{g_1(\theta,x)},$$

where g_1 is differentiable and g_2 is lower semi-continuous, and $\imath_{\mathcal{K}}$ is the convex indicator of $\mathcal{K}$ defined by $\imath_{\mathcal{K}}(x) = 0$ if $x \in \mathcal{K}$ and $\imath_{\mathcal{K}}(x) = \infty$ otherwise.

In both cases we have that

$$g_1(\theta, x) = \sum_{j=1}^{d_y} \left(y_j \log(s(v_j^T x)) + (1 - y_j) \log(s(-v_j^T x)) \right) + C$$

where C is a constant. As shown in Akyildiz et al. [2025, Section 6.1.1], the function g_1 is gradient Lipschitz and strictly convex but not strongly convex. The function g_2 satisfies **A1** for both the Laplace and the uniform prior, as observed in Pereyra [2016], in the case of the Laplace prior g_2 also satisfies **A3** while the uniform prior does not lead to a Lipschitz g_2. Since g_1 does not depend on θ, **A5** holds for the Laplace prior.

Dataset. We create a synthetic dataset by first fixing the value of θ and sampling the latent variable $x \in \mathbb{R}^{50}$ from the corresponding prior. We then sample the 900 observations from a Bernoulli distribution with parameter $s(v_j^T x)$, where s is the logistic function and the entries of the covariates v_j are drawn from a uniform distribution $\mathcal{U}(-1, 1)$. The true value of θ is set to $\bar{\theta}_\star = -4$ for the Laplace prior and $\bar{\theta}_\star = 1.5$ for the uniform one.

Implementation details. The x-gradients of g_1 can be computed analytically. To choose the optimal values of γ and λ for the different implementations, we perform a grid search in the range $[5 \times 10^{-4}, 0.5]$. The selected optimal values are displayed in Table 5. We note that in PIPGLA the optimal values for λ, γ turn out to be when $\lambda = \gamma$.

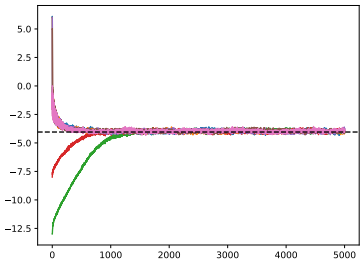
Table 5: Optimal hyperparameters for Bayesian logistic regression example. Recall that for PPGD and PIPULA we only have the γ parameter since we set $\lambda = \gamma$.

Algorithm	Approx./Iterative	γ		λ	
		Laplace	Unif	Laplace	Unif
PPGD	Approx	0.1	0.03	–	–
	Iterative	0.06	–	–	–
PIPULA	Approx	0.06	0.03	–	–
	Iterative	0.06	–	–	–
MYPGD	Approx	0.05	0.001	0.25	0.01
	Iterative	0.05	–	0.005	–
MYIPLA	Approx	0.05	0.001	0.35	0.01
	Iterative	0.05	–	0.005	–
PIPGLA	Approx	0.01	0.02	0.01	0.02
	Iterative	0.01	–	0.01	–

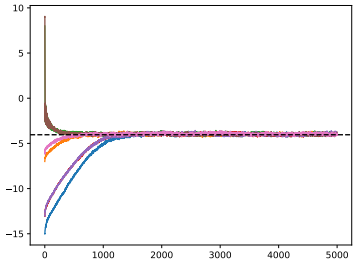
Results. Table 6 extends the results in Table 2 by also including the results for PPGD, PIPULA and IPLA (as a benchmark). Figure 4 shows the θ-iterates obtained with MYIPLA and PIPGLA starting from 7 different initial values θ_0 and using the approximate solver for $\text{prox}_{g_2}^{\lambda}$ with $g_2(\theta, x) = \sum_{i=1}^{d_x} |x_i - \theta|$ and an iterative procedure using 40 iterations in each step. We observe that the iterative solver results in a slightly slower convergence to stationarity, but overall the two sets of algorithms converge to the same true value of θ. We also observe that the convergence to stationarity for PIPGLA is much slower compared to MYIPLA. However, if we increase the value of γ in the hope of faster convergence, the iterates either do not converge to the true value or the standard deviation is significantly larger. For all algorithms considered, approximate solvers are 25% faster than iterative solvers (see Table 6).

We also compare the results for the uniform prior, in this case we only use the approximate proximity map (Figure 5), as the iterative approach is not numerically stable.

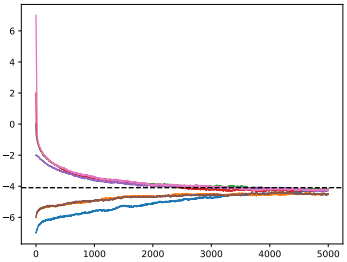
Since all the algorithms considered aim at estimating the parameter θ by sampling from a distribution which concentrates around $\bar{\theta}_\star$, we compare the estimators of $\bar{\theta}_\star$ obtained by using only the last iterate θ_{K+1}^N and averaging over a number of iterates. We compare the normalised MSE (NMSE) for θ for the estimator obtained by averaging the θ-iterates after discarding a burn-in of 1500 samples (column named *avg*) against using the last θ of the chain (column *last*). The results are in agreement, with the NMSE for the averaged estimator having lower variance in most settings (Table 7).



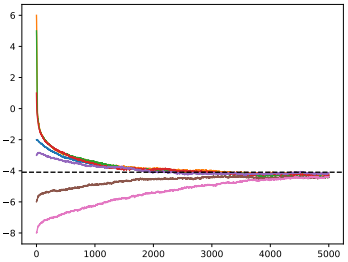
(a) MYIPLA, approximated $\mathrm{prox}_{g_2}^{\lambda}$.



(b) MYIPLA, iterative solver for $\mathrm{prox}_{g_2}^{\lambda}$.

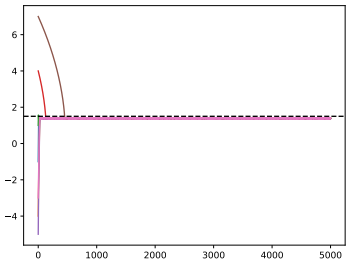


(c) PIPGLA, approximated $\mathrm{prox}_{g_2}^{\lambda}$.

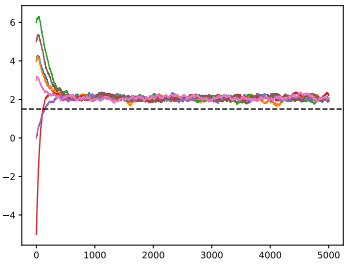


(d) PIPGLA, iterative solver for $\mathrm{prox}_{g_2}^{\lambda}$.

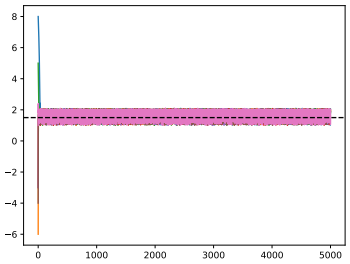
Figure 4: Bayesian logistic regression with isotropic Laplace priors on the regression weights $\prod_i \mathrm{Laplace}(x_i|\theta, 1)$, with true $\theta = -4$. Each plot shows the θ-iterates for 7 different starting points.



(a) MYPGD.



(b) MYIPLA.



(c) PIPGLA.

Figure 5: Bayesian logistic regression with isotropic uniform priors on the regression weights $\prod_i \mathcal{U}(x_i| - \theta, \theta)$, with true $\theta = 1.5$. The plot displays the θ-iterates for 7 randomly chosen starting points.

Table 6: Bayesian logistic regression for Laplace and uniform priors. Normalised MSE (NMSE) for θ for different algorithm when run 500 times using 50 particles, 5000 steps and different starting points. Computation times and NMSEs are averaged over the 500 replicates. The second column indicates whether the proximal map is calculated approximately or iteratively, using 40 steps in each iteration. For the uniform prior case we have not implemented the iterative method.

Algorithm	Approx./Iterative	NMSE (%)		Times (s)	
		Laplace	Unif	Laplace	Unif
PPGD	Approx	14.70 ± 4.42	3.63 ± 4.93	102.6 ± 5.1	107.9 ± 5.5
	Iterative	19.04 ± 1.34	–	122.3 ± 5.1	–
PIPULA	Approx	12.18 ± 1.62	4.71 ± 6.02	98.8 ± 5.7	101.0 ± 4.0
	Iterative	19.22 ± 1.28	–	126.2 ± 3.8	–
MYPGD	Approx	6.09 ± 0.34	$\mathbf{0.60 \pm 0.23}$	91.9 ± 4.8	109.3 ± 4.6
	Iterative	4.44 ± 1.40	–	129.7 ± 15.8	–
MYIPLA	Approx	4.42 ± 1.32	15.26 ± 4.44	89.9 ± 4.2	97.0 ± 4.2
	Iterative	4.67 ± 1.60	–	120.5 ± 10.1	–
PIPGLA	Approx	2.30 ± 0.58	6.83 ± 3.97	116.5 ± 5.5	103.1 ± 8.0
	Iterative	$\mathbf{2.02 \pm 0.54}$	–	122.9 ± 6.9	–
IPLA	–	7.76 ± 3.39	20.12 ± 2.88	$\mathbf{81.1 \pm 3.0}$	$\mathbf{82.9 \pm 4.9}$

E.3 BAYESIAN NEURAL NETWORK

E.3.1 Sparsity Inducing Prior: MNIST

Our setting is equivalent to assuming that the datapoints' labels l are conditionally independent given the features f and network weights $x = (w, v)$, and therefore have the following probability density

$$p(l|f, x) \propto \exp\Big(\sum_{j=1}^{40} v_{lj} \tanh\Big(\sum_{i=1}^{784} w_{ji} f_i\Big)\Big).$$

We assign priors $p_\alpha(w) = \prod_i \text{Laplace}(w_i|0, e^{2\alpha})$ and $p_\beta(v) = \prod_i \text{Laplace}(v_i|0, e^{2\beta})$ to the input and output layer's weights, respectively, and learn $\theta = (\alpha, \beta)$ from the data. The model's density is given by

$$p_\theta(x, \mathcal{Y}_{\text{train}}) = \prod_i \text{Laplace}(w_i|0, e^{2\alpha}) \prod_j \text{Laplace}(v_j|0, e^{2\beta}) \prod_{(f,l)\in\mathcal{Y}_{\text{train}}} p(l|f, x),$$

where x denotes the weight matrices, i.e. $x = (w, v)$. We note that the log density can be decomposed as

$$-\log p_\theta(x, \mathcal{Y}_{\text{train}}) = \underbrace{d_w\alpha + \sum_i |w_i|e^{-2\alpha} + d_v\beta + \sum_j |v_j|e^{-2\beta}}_{g_2(\theta,x)} \underbrace{- \sum_{(f,l)\in\mathcal{Y}_{\text{train}}} \log p(l|f, x)}_{g_1(\theta,x)},$$

where d_w and d_v denote the dimensions of the weights w and v, respectively, g_1 is differentiable and does not depend on θ and g_2 is proper, convex and lower semi-continuous. We have derived an approximation to the proximity map of g_2 in E.1.2.

Dataset. We use the MNIST dataset. Features are normalised so that each pixel has mean zero and unit standard deviation across the dataset. We split the dataset into 80/20 training and test sets.

Proximal operator of g_2. As g_2 can be expressed as $g_2(\theta, x) = g_2(\alpha, w) + g_2(\beta, v)$, we can compute their proximal operators separately. It is sufficient to calculate the proximal operator for $g_2(w, \alpha)$ since it is equivalent to that of $g_2(v, \beta)$. To do so, we have that

$$\text{prox}^\lambda_{g_2}(\alpha, w) = \underset{(u_0,u)}{\arg\min}\, h(u_0, u), \quad h(u_0, u) = u_0 d_w + \sum_i |u_i|e^{-2u_0} + \|(u_0, u) - (\alpha, w)\|^2/(2\lambda),$$

whose approximate solution is calculated in Section E.1.2.

Table 7: Bayesian logistic regression for Laplace and uniform priors. Normalised MSE (NMSE) for the last iterate of θ (*last*) and the posterior mean after discarding a burn-in of 1500 samples (*avg*). Each different algorithm is run 500 times for different starting points using 50 particles and 5000 steps. NMSEs are averaged over the 500 replicates. The second column indicates whether the proximal map is calculated approximately or iteratively, using 40 steps in each iteration. For the uniform prior case, we did not implement the iterative method due to numerical instabilities.

Algorithm	Approx/	Laplace		Uniform	
	Iterative	NMSE last(%)	NMSE avg(%)	NMSE last(%)	NMSE avg(%)
PPGD	Approx	14.70 ± 4.42	16.73 ± 0.83	3.63 ± 4.93	$\mathbf{0.11 \pm 0.04}$
	Iterative	19.04 ± 1.34	18.66 ± 0.60	$-$	$-$
PIPULA	Approx	12.18 ± 1.62	12.34 ± 0.82	4.71 ± 6.02	0.12 ± 0.01
	Iterative	19.22 ± 1.28	18.63 ± 0.79	$-$	$-$
MYPGD	Approx	6.09 ± 0.34	4.94 ± 0.51	$\mathbf{0.60 \pm 0.23}$	0.60 ± 0.02
	Iterative	4.44 ± 1.40	4.33 ± 0.59	$-$	$-$
MYIPLA	Approx	4.42 ± 1.32	4.31 ± 0.67	15.26 ± 4.44	16.01 ± 2.01
	Iterative	4.67 ± 1.60	4.45 ± 0.42	$-$	$-$
PIPGLA	Approx	2.30 ± 0.58	2.45 ± 0.94	6.83 ± 3.97	4.22 ± 0.07
	Iterative	$\mathbf{2.02 \pm 0.54}$	$\mathbf{2.03 \pm 0.88}$	$-$	$-$

Implementation details. For the x-gradients of g_1, we use JAX's grad function (implementing a version of autograd). Plugging the expressions above in the corresponding equations, we can implement the proposed algorithms. However, due to the high dimensionality of the latent variables, we stabilise the algorithm using the heuristics discussed in Section 2 of Kuntz et al. [2023]. This simply entails dividing the gradients and proximal mapping terms of the updates of α and β by d_w and d_v. We then set $\gamma = 0.05$ and $\lambda = 0.5$ (after performing a grid search) which ensures that the algorithms are not close to losing stability. In addition, the weights of the network are initialised according to the assumed prior. This is done by setting each weight to $\pm\, a \log u$ where $u \sim \mathcal{U}(0,1)$, the sign is chosen uniformly at random and $a > 0$ is interpreted as the average initial size of the weights. Williams [1995] suggests setting $a = 1/\sqrt{2m}$ for w and $a = 1.6/\sqrt{2m}$ for v, where m is the fan-in of the destination unit.

Predictive performance metrics. To allow comparison, we use the same performance metrics as in Kuntz et al. [2023]. We include their presentation of this metrics for completeness.

Given a new feature vector $\hat{f}$, the posterior predictive distribution for a label $\hat{l}$ associated with the marginal likelihood maximiser $\bar{\theta}_\star$ is given by

$$p_{\bar{\theta}_\star}(\hat{l}|\hat{f}, \mathcal{Y}_{\text{train}}) = \int p(\hat{l}|\hat{f}, x) p_{\bar{\theta}_\star}(x|\mathcal{Y}_{\text{train}}) \mathrm{d}x.$$

As $p_{\bar{\theta}_\star}(x|\mathcal{Y}_{\text{train}})$ is unknown, we approximate it with the empirical distribution of the final particle cloud $q = N^{-1} \sum_{i=1}^N \delta_{X_K^i}$, leading to

$$p_{\bar{\theta}_\star}(\hat{l}|\hat{f}, \mathcal{Y}_{\text{train}}) \approx \int p(\hat{l}|\hat{f}, x) q(\mathrm{d}x) = \frac{1}{N} \sum_{i=1}^N p(\hat{l}|\hat{f}, X_K^i) =: g(\hat{l}|\hat{f}).$$

The metrics considered to evaluate the approximation of the predictive power are the average classification error over the test set $\mathcal{Y}_{\text{test}}$, i.e.

$$\text{Error} := \frac{1}{|\mathcal{Y}_{\text{test}}|} \sum_{(f,l) \in \mathcal{Y}_{\text{test}}} \mathbb{1}\{l = \hat{l}(f)\}, \quad \text{where } \hat{l}(f) := \arg\max_{\hat{l}} g(\hat{l}|\hat{f}),$$

and the log pointwise predictive density (LPPD, Vehtari et al. [2017])

$$\text{LPPD} := \frac{1}{|\mathcal{Y}_{\text{test}}|} \sum_{(f,l) \in \mathcal{Y}_{\text{test}}} \log(g(l|f)).$$

Under the assumption that data is drawn independently from $p(l, f)$, we have the following approximation for large test data

Table 8: Bayesian neural network. Performance of BNN with Laplace (implemented using MYIPLA) and Normal priors (implementation with PGD) when setting weights from the final particle cloud below a certain threshold to zero. The second column refers to whether the particles are averaged before (✓) or after (✗) calculating the performance.

Prior	Average over particles?	% of zero weights		Thresholds		Error (%)	LPPD
		Layer 1	Layer 2	Layer 1	Layer 2		
Laplace	✓	74	48	0.2	0.2	**7.0**	**−0.23**
	✗	56	35	1	1	**1.5**	**−0.07**
Normal	✓	74	48	0.5	1.1	15	−0.74
	✓	16	15	0.2	0.2	16	−0.78
	✗	56	35	7	4	2.0	−0.11
	✗	8.6	7.1	1	1	1.5	−0.10

sets,

$$\begin{aligned}\text{LPPD} &\approx \int \log(g(l|f))p(\mathrm{d}l,\mathrm{d}f) = \int \left[\int \log\left(\frac{g(l|f)}{p(l|f)}\right)p(\mathrm{d}l|\mathrm{d}f)\right]p(\mathrm{d}f) + \int \log(p(l|f))p(\mathrm{d}l,\mathrm{d}f)\\ &= -\int \mathrm{KL}(g(\cdot|f)\|p(\cdot|f))p(\mathrm{d}f) + \int \log(p(l|f))p(\mathrm{d}l,\mathrm{d}f).\end{aligned}$$

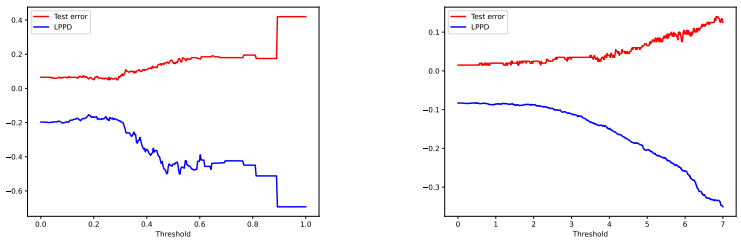
This means that the larger the LPPD is, the smaller the mean KL divergence between our classifier $g(l|f)$ and the optimal classifier $p(l|f)$.

Results. First, it is important to discuss whether the Laplace prior is more appropriate in this setting than the Normal one. Jaynes [1968] provides two reasons why the Laplace prior is particularly suitable for Bayesian neural network models. Firstly, for any feedforward network there is a functionally equivalent network in which the weight of a non-direct connection has the same size but opposite sign, therefore consistency demands that the prior for a given weight w is a function of $|w|$ alone. Secondly, if it is assumed that all that is known about $|w|$ is its scale, and that the scale of a positive quantity is determined by its mean rather than some higher order moment, then the maximum entropy distribution for a positive quantity constrained to a given mean is the exponential distribution. It would follow that the signed weight w has a Laplace density [Williams, 1995].

We have examined the sparsity-inducing nature of the Laplace prior versus a normal one in Figure 2 and Table 3. As mentioned in the main text, the sparse representation of our experiment also has the advantage of producing models that are smaller in terms of memory usage when small weights are zeroed out. To investigate this, we set to zero all weights below a certain threshold and analyse the performance of the compressed weight matrices. We consider two cases, averaging the particles of the final cloud $X_{500}^1, \ldots, X_{500}^{100}$, applying the threshold and then calculating the performance, and secondly, setting to zero small values of each particle of the cloud and averaging the performance of each particle. We compare the results for the Bayesian neural networks with Laplace and Normal priors (Table 8). It is important to note that when applying the same threshold to both cases, the Laplace prior leads to a very compressed weight matrix compared to the Normal prior, i.e. there is a significant difference in the percentage of weights set to zero. We observe that when setting the same proportion of weights to zero in both layers, the performance of the BNN with Laplace priors is better in terms of the log pointwise predictive density than that of the BNN with Normal priors, especially when averaging the final cloud of particles before computing the performance.

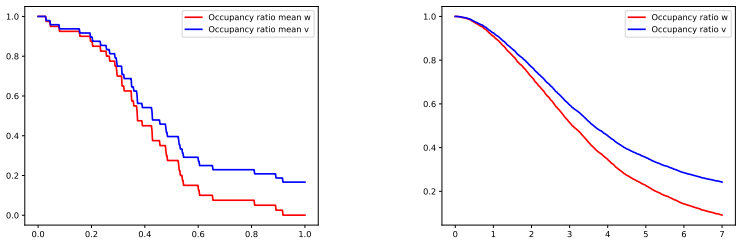
Figure 6 shows how the performance metrics evolve when weights below a certain threshold are set to zero, when particles are averaged before (6a) or after (6b) computing the performance for MYIPLA.

Once we have set the weights of the matrix below a certain threshold to zero, it is necessary to explore the dead units. These are hidden units all of whose input or output weights are zero [Williams, 1995]. In both cases, the unit is redundant and it can be eliminated to obtain a functionally equivalent network architecture, we will called this new effective weight matrix w_{pruned}. The occupancy ratio of a weight matrix w [Marinó et al., 2023] is defined as $\psi = \text{size}(w_{\text{pruned}})/\text{size}(w)$, where size denotes the memory size. The inverse of ψ is the compression ratio. We compute the occupancy ratio of the weight matrix for both the hidden and output layer for different values of the pruning threshold. We do this for each particle of the final cloud and obtain the average as well as for the averaged final particle cloud, results are shown in Figure 7.



(a) (b)

Figure 6: Evolution of the performance metrics when weights below a certain threshold are set to zero, when particles are averaged before (a) or after (b) computing the performance.



(a) (b)

Figure 7: Occupancy ratio for the weights matrices of the hidden and output layers as a function of the pruning threshold, when particles are averaged before (a) or after (b) computing the occupancy ratio.

E.3.2 Sparsity Inducing Prior: CIFAR10

We further evaluate our methods on a classification task using a more complex dataset: CIFAR10. As with the MNIST dataset, to reduce the cost of computing the gradients on a big dataset, we subsample 5000 data points with labels *plane*, *car*, *ship* and *truck*.

Given that the data consists of colour images, we employ a convolutional neural network (CNN) architecture. Specifically, we use a combination of convolutional layers, max pooling layers, and linear layers with non-linear activation functions. For simplicity, we apply a sparsity-inducing prior only to the linear layers, and not to the convolutional ones. The sparsity inducing prior for each layer with weight matrix w is given by $p_\alpha(w) = \prod_i \text{Laplace}(w_i|0, e^{2\alpha})$ where α is learn from the data. The network structure and layer dimensions are as follows.

- Convolutional layer (Deterministic): Conv2d(3, 6, 5)
- Max pooling layer (Deterministic): MaxPool2d(2, 2)
- Convolutional layer (Deterministic): Conv2d(6, 16, 5)
- Linear layer with sparsity inducing prior + SELU activation function: Linear($16 \times 5 \times 5$, 512)
- Linear layer with sparsity inducing prior + SELU activation function: Linear(512, 256)
- Linear layer with sparsity inducing prior + SELU activation function: Linear(256, 128)
- Linear layer with sparsity inducing prior: Linear(128, 4)

Table 9 presents quantitative results for the variance of the weights and error metrics. The last column provides a measure of the sparsity-inducing effect of the Laplace prior on the linear layers.

E.3.3 Non-Differentiable Activation Functions

During gradient checking in neural network training, a potential source of inaccuracy arises from the presence of non-differentiable points in the objective function [Kumar, 2024]. These non-smooth points often result from the use of activation

Table 9: Bayesian neural network on CIFAR10 dataset. Test errors and log pointwise predictive density (LPPD) achieved using the final particle cloud with $N = 50$. Computation times and standard deviation of the empirical distribution of the weight matrix w for linear layers are also provided.

Algorithm	Error (%)	LPPD ($\times 10^{-1}$)	Time (s)	Std. w
MYPGD	5.27 ± 0.95	-4.41 ± 0.38	201	3.10
MYIPLA	$\mathbf{5.23 \pm 1.31}$	-5.05 ± 0.45	199	3.22
PIPGLA	5.39 ± 1.02	$\mathbf{-4.32 \pm 0.37}$	295	**2.85**
PGD	6.01 ± 1.15	-5.73 ± 0.40	**178**	11.51
SOUL	9.11 ± 2.03	-7.68 ± 1.56	433	15.68
IPLA	5.40 ± 1.33	-5.90 ± 0.75	181	15.73

functions such as the Rectified Linear Unit (ReLU), defined as $\max(0, x)$, as well as from the hinge loss in support vector machines, maxout neurons, among others. To give a concrete example, consider the ReLU activation function and $x < 0$ but very close to 0. The analytic gradient evaluated at x is equal to 0. However, the numerical gradient can be non-zero when using a finite difference approximation in case $x + h > 0$.

Our method provides a principled way of dealing with these non-differentiable points. To illustrate this, we present a simple example similar to the one in the previous section. Here, we consider a Bayesian neural network with a Normal prior distribution on the weights to classify MNIST digits 1 and 7, instead of 4 and 9. Additionally, we use a linear approximation of $tanh$ as the activation function to mitigate the dying neuron problem associated with ReLU [Lu et al., 2020], while noting that it still remains non-differentiable. This linear approximation is defined as

$$h(x) = \begin{cases} -1 & \text{if } x < -1, \\ x & \text{if } x \in [-1, 1], \\ 1 & \text{if } x > 1. \end{cases}$$

Furthermore, we can compute the proximal mapping of h which is given by

$$\text{prox}_h^\lambda(x) = \begin{cases} x & \text{if } x < -1, \\ -1 & \text{if } x \in [-1, -1 + \lambda], \\ x - \lambda & \text{for } x \in [-1 + \lambda, 1 - \lambda], \\ 1 & \text{for } x \in [1 - \lambda, 1], \\ x & \text{if } x > 1, \end{cases} \tag{53}$$

where we have applied the first order optimality condition and used the subgradient of the function at $x = -1$ and 1 which is given by the sets $[0, 1]$ and $[-1, 0]$, respectively. Therefore, in this setting we have the following likelihood

$$p(l|f, x) \propto \exp\Big(\sum_{j=1}^{40} v_{lj} h_{LR}\Big(\sum_{i=1}^{784} w_{ji} f_i\Big)\Big).$$

We assign priors $p_\alpha(w) = \prod_i \mathcal{N}(w_i|0, e^{2\alpha})$ and $p_\beta(v) = \prod_i \mathcal{N}(v_i|0, e^{2\beta})$ to the input and output layer's weights, respectively, and learn $\theta = (\alpha, \beta)$ from the data. Hence, model's density is given by

$$p_\theta(x, \mathcal{Y}_{\text{train}}) = \prod_i \mathcal{N}(w_i|0, e^{2\alpha}) \prod_j \mathcal{N}(v_j|0, e^{2\beta}) \prod_{(f,l)\in\mathcal{Y}_{\text{train}}} p(l|f, x),$$

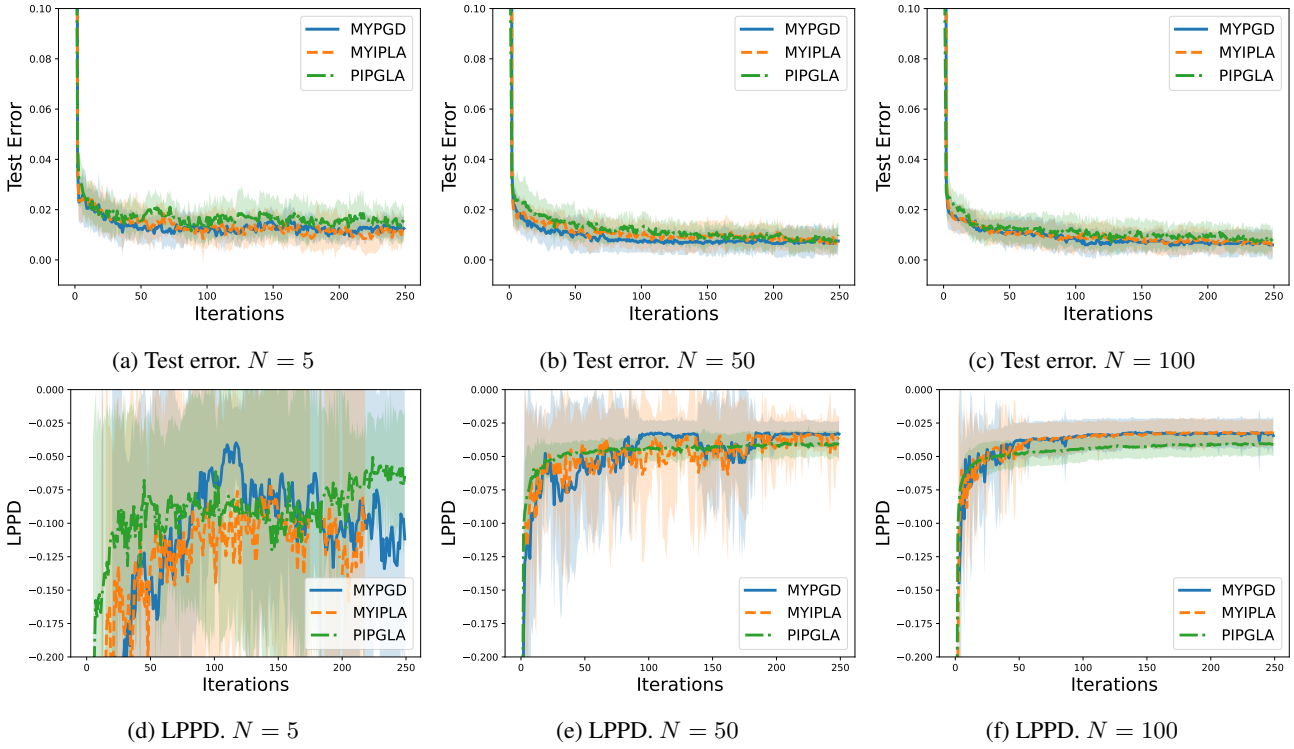
where x denotes the weight matrices, i.e. $x = (w, v)$. We note that the log density can be decomposed as

$$-\log p_\theta(x, \mathcal{Y}_{\text{train}}) = \underbrace{d_w\alpha + \frac{1}{2}\sum_i |w_i|^2 e^{-2\alpha} + d_v\beta + \frac{1}{2}\sum_j |v_j|^2 e^{-2\beta}}_{g_1(\theta,x)} \underbrace{- \sum_{(f,l)\in\mathcal{Y}_{\text{train}}} \log p(l|f, x)}_{g_2(\theta,x)},$$

where d_w and d_v denote the dimensions of the weights w and v, respectively, g_1 is differentiable and depends on θ and x, while g_2 is proper, convex and lower semi-continuous and only depends on x, that is, $g_2(\theta, x) = g_2(x)$. As a result, the

Table 10: Bayesian neural network with non-differentiable activation function. Test errors and log pointwise predictive density (LPPD) achieved using the final particle cloud with $N = 50$ and 500 iterations.

Algorithm	Error (%)	LPPD $\left(\times 10^{-2}\right)$	Times (s)
MYPGD	0.75 ± 0.68	$\mathbf{-3.36 \pm 1.18}$	**40**
MYIPLA	$\mathbf{0.70 \pm 0.50}$	-4.28 ± 2.86	**40**
PIPGLA	0.90 ± 0.49	-3.76 ± 0.96	68



(a) Test error. $N = 5$ (b) Test error. $N = 50$ (c) Test error. $N = 100$

(d) LPPD. $N = 5$ (e) LPPD. $N = 50$ (f) LPPD. $N = 100$

Figure 8: Evolution of the classification error on a test set (top) and the log pointwise predictive density (LPPD) (bottom) over iterations in the BNN experiment with non-differentiable activation function, using 250 iterations. Values averaged over 100 runs.

non-differentiability affects only the latent variables x. In this case, we can compute the proximity map of g_2 by using the expression for the proximity map of the activation function, provided in (53).

We follow the same implementation details as outlined in the previous section and use the same performance metrics: average classification error over a test set and log pointwise predictive density. The results for the proposed proximal algorithms are provided in Table 10 together with the computation times for $N = 50$ and 500 iterations. In addition, plots of the evolution of the different performance metrics for different number of particles are shown in Figure 8. We observe that the standard deviation of the LPPD across runs decreases as the number of particles increases. Moreover, PIPGLA exhibits a lower standard deviation compared to the other methods.

E.4 IMAGE DEBLURRING

We consider the problem of recovering a high-quality image from a blurred and noisy observation $y = Hx + \varepsilon$, where H is a blurring operator that blurs a pixel $x_{i,j}$ uniformly with its closest neighbours (10×10 patch), and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. The log prior is proportional to the total variation defined as $TV(x) = \|\nabla_d x\|_1$, where $\|\cdot\|_1$ is the ℓ_1 norm and ∇_d is the two-dimensional discrete gradient operator, which is non-differentiable. The proportionality parameter, e^θ, which controls the strength of this log prior, typically requires manual tuning. Instead of fixing this parameter manually, we estimate its optimal value using our proposed algorithms. Note that we exponentiate θ to ensure its positivity.

The posterior distribution for the model takes the form

$$p_\theta(y|x) \propto \exp\left(-\|y - Hx\|^2/(2\sigma^2) - e^\theta TV(x) + \log C(\theta)\right),$$

where $C(\theta)$ is proportional to the normalising constant of the prior distribution. To compute $C(\theta)$, we start by considering the case when $\theta = 0$. In this case, the total variation prior is given by

$$p(x) = C\exp(-TV(x)),$$

where C is constant. For $\theta \neq 0$, the prior $p_\theta(x)$ can be expressed using the pushforward measure as

$$p_\theta(x) = T_{e^\theta}\#p(x) = e^{d_x\theta}p(e^\theta x),$$

where $T_{e^\theta}\#$ denotes the pushforward operator and d_x is the dimension of x. Due to the linearity of the total variation norm, it follows that

$$p_\theta(x) = Ce^{d_x\theta}\exp\left(-e^\theta TV(x)\right).$$

Thus, we obtain that theta $C(\theta) = e^{d_x\theta}$. For the experiments, we employ the algorithms proposed by [Douglas and Rachford, 1956] and [Chambolle, 2004] to efficiently compute the proximal operator of the total variation norm. Due to the difficulty of computing the joint proximal operator over the parameter θ and latent variables x, we have consider hybrid versions of the algorithms, which use standard gradient-based updates for the parameters and proximal updates for the particles. That is, the updates for the hybrid MYIPLA algorithm are given by

$$\begin{aligned}
\theta_{n+1}^N =& \theta_n^N - \frac{\gamma}{d_x N}\sum_{i=1}^N e^{\theta_n^N}TV(X_n^{i,N}) + \gamma + \sqrt{\frac{2\gamma}{N}}\xi_{n+1}^{0,N},\\
X_{n+1}^{i,N} =& \Big(1 - \frac{\gamma}{\lambda}\Big)X_n^{i,N} - \gamma\frac{H^\intercal(HX_n^{i,N} - y)}{\sigma^2} + \frac{\gamma}{\lambda}\operatorname{prox}^\lambda_{e^{\theta_n^N}TV}(X_n^{i,N}) + \sqrt{2\gamma}\,\xi_{n+1}^{i,N}.
\end{aligned}$$

Note that as in the Bayesian neural network example, we apply the heuristic of dividing the gradient term in the θ updates by d_x for numerical stability. For PIPGLA, the update for the parameter θ remains the same, while the updates for the particles are of the form

$$\begin{aligned}
X_{n+1/2}^{i,N} &= X_n^{i,N} - \gamma\frac{H^\intercal(HX_n^{i,N} - y)}{\sigma^2} + \sqrt{2\gamma}\,\xi_{n+1}^{i,N},\\
X_{n+1}^{i,N} &= \operatorname{prox}^\lambda_{e^{\theta_{n+1}^N}TV}\big(X_{n+1/2}^{i,N}\big).
\end{aligned}$$

Analogous forms are defined for the proximal PGD algorithms.

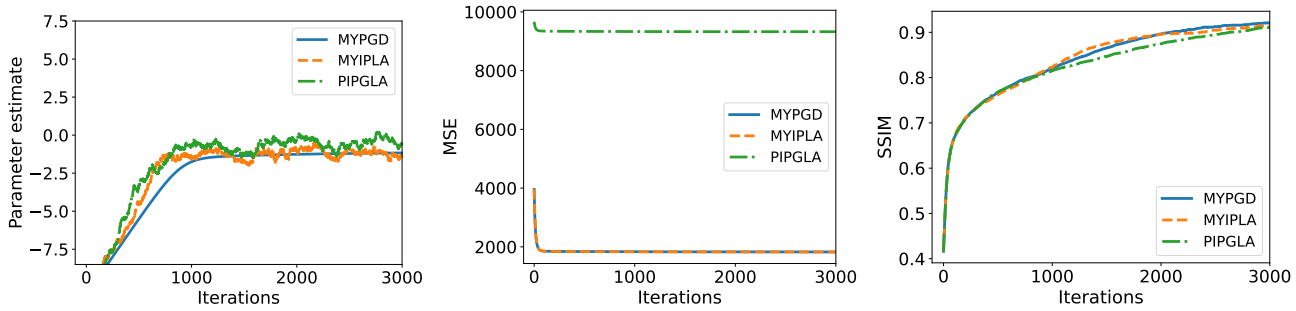
Dataset. We use black and white images with pixels values ranging from 0 to 255. The dimensions of the acoustic guitar image are $d_x = n_1 \times n_2 = 584 \times 238$, while the dimensions of the boat image (a standard benchmark in the image reconstruction literature) are $d_x = 512 \times 512$.

Implementation details. We implement the proximal operator of the total variation using the *proxTV* Python package [Barbero and Sra, 2011, 2018]. Specifically, we employ the Douglas-Rachford method introduced by Douglas and Rachford [1956] and the Chambolle-Pock method [Chambolle, 2004]. The Douglas-Rachford method is significantly faster than the Chambolle-Pock method. It is important to note that increasing the precision of the Moreau-Yosida envelope significantly slows down the computation of the proximal operator when using these numerical schemes. We set $\gamma = 0.01$, and $\lambda = 0.4$ for MYPGD and MYIPLA and $\lambda = 0.001$ for PIPGLA (after performing a grid search) which ensures that the algorithms are not close to losing stability. In addition, the pixels of the initial particles are drawn from a normal distribution with mean $\mu = 50$ and scale parameter 10, while the initial parameter estimate θ_0 is sampled from a uniform distribution over $[-15, 10]$.

Performance metrics. To evaluate the performance of our algorithms in image reconstruction, we evaluate the mean squared error (MSE) and the structural similarity index (SSIM) between the particle cloud and the ground-truth image. The SSIM quantifies image quality by comparing luminance, contrast, and structural details.

(a) Original (b) Blurred (c) MYPGD (d) MYIPLA (e) PIPGLA

Figure 9: Image deblurring experiment. All the algorithms use $N = 10$ particles and are run for 3000 iterations with a burn-in of 100 iterations.



(a) θ estimates (b) MSE (c) Structural similarity index (SSIM)

Figure 10: Evolution of different quantities over iterations in the image deblurring experiment with $N = 10$ particles for the acoustic guitar image. The plots are shown after discarding a burn-in period of 100 iterations and the initial parameter is $\theta_0 = -10.5$.

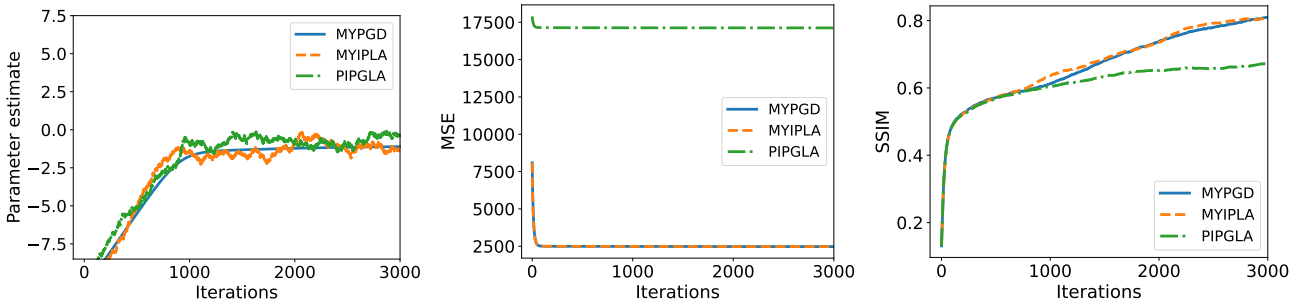
Results. Figures 9 and 11 display the original and blurred images alongside the reconstructed images obtained using our different proximal algorithms. The methods are run for 3000 iterations (with a burn-in of 100 iterations) and $N = 10$ particles, employing the Douglas-Rachford method to numerically evaluate the proximal operator of the total variation norm. Figures 10 and 12 illustrate the evolution of the parameter estimates θ, the mean squared error and the SSIM, after discarding a burn-in period of 100 iterations. The high MSE for PIPGLA (Figures 10b and 12b) arises from the difference in the shades of grey between the reconstructed and the original images, remaining large regardless of the choice of the proximal parameter λ. Besides, the optimal value for the strength of the total variation prior achieved by the algorithms, $e^{\theta} \approx 0.35$ (for both test images) is close to the value set manually in similar works for image reconstruction (e.g. Durmus et al. [2018], Goldman et al. [2022], Pereyra [2016]).

E.5 NUCLEAR-NORM MODELS FOR LOW RANK MATRIX ESTIMATION

In this section, we demonstrate another application of our methods: the problem of matrix completion. Matrix completion [Candès and Plan, 2010, Liu et al., 2018] focuses on recovering an intact matrix with low-rank property from incomplete data. Its application varies from wireless communications [Kortas et al., 2017], traffic sensing [Mardani and Giannakis, 2014] to

(a) Original (b) Blurred (c) MYPGD (d) MYIPLA (e) PIPGLA

Figure 11: Image deblurring experiment. All the algorithms use $N = 10$ particles and are run for 3000 iterations with a burn-in of 100 iterations.



(a) θ estimates (b) MSE (c) Structural similarity index (SSIM)

Figure 12: Evolution of different quantities over iterations in the image deblurring experiment with $N = 10$ particles for the boat image. The plots are shown after discarding a burn-in period of 100 iterations and the initial parameter is $\theta_0 = -8.1$.

integrated radar and recommender systems [Gogna and Majumdar, 2015]. The low-rank prior knowledge is incorporated in the model using the nuclear-norm of the matrix [Fazel, 2002]. However, similar to the image deblurring example, the strength of this prior is a hyperparameter that must be set manually. Instead, we estimate the optimal value of this parameter, thereby extending the applicability of proximal methods that typically perform MLE, rather than MMLE, as in our algorithms.

We conduct a graphical posterior predictive check of the widely used nuclear norm model for low-rank matrices, similar to the example in Pereyra [2016], but in the context of matrix completion rather than matrix denoising. Let x be an unknown low-rank matrix of size $n_1 \times n_2$. Consider a mask M_Ω, where Ω is a set of indices from a matrix of size $n_1 \times n_2$. When the mask is applied to the matrix x, i.e., $M_\Omega X$, only the entries of the matrix corresponding to indices in Ω are observed. Furthermore, after the masking operation, we do not have direct access to the observed entries but instead observe a noisy version of them, where the observational noise has mean zero and covariance $\sigma^2 I$. Thus, our observations are given by $y = M_\Omega x + \sigma^2 \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, I)$. It is important to highlight, that we will also estimate with our algorithms the scale parameter σ, rather than requiring it to be fixed manually.

Our objective is to recover x from y under the prior knowledge that x has low rank, that is, most of its singular values are zero. A convenient model for this type of problem is the nuclear norm prior, which is a sparsity-inducing prior, given by

$$p_\theta(x) = C(\theta_1) e^{-e^{\theta_1} \|x\|_{\text{tr}}},$$

where $\|\cdot\|_{\text{tr}}$ is the trace (or nuclear) norm, which is a convex envelope of the rank function [Bach, 2008], and is defined as

$$\|x\|_{\text{tr}} = \sum_{i=1}^{r} \sigma_i(x),$$

$r = \text{rank}(x)$ and $\sigma_1(x) \geq \cdots \geq \sigma_r(x) \geq 0$ are the singular values. Besides, the constant $C(\theta_1)$ can be computed using the pushforward argument, as in Section E.4, leveraging the linearity of the trace norm. Specifically, it is given by $C(\theta_1) = C e^{d_x \theta_1}$, where d_x denotes the dimension of the original matrix x and C is a constant. The posterior distribution of our model can be written as

$$p_\theta(y|x) \propto \frac{e^{d_x \theta_1}}{e^{d_y \theta_2}} \exp\left(-\frac{\|M_\Omega x - y\|^2}{2e^{2\theta_2}} - e^{\theta_1} \|x\|_{\text{tr}} \right).$$

Therefore, the negative log density can be decomposed as

$$U(\theta, x) = \underbrace{-d_x\theta_1 + d_y\theta_2 + \frac{\|M_\Omega x - y\|^2}{2e^{2\theta_2}}}_{g_1(\theta,x)} + \underbrace{e^{\theta_1}\|x\|_{\text{tr}}}_{g_2(\theta,X)},$$

where d_y denotes the number of observed entries. Note that we exponentiate the parameters θ_1 and θ_2 to ensure their positivity.

Dataset. We use the *checkerboard* image of size 188×188 and rank 2. We add Gaussian observational noise with variance $\sigma^2 = 0.1$ and mask 30% of the pixels in the image.

Proximal operator of g_2. Recall that $g_2(\theta, x)$ is of the form

$$g_2(\theta, x) = e^{\theta_1}\|x\|_{\text{tr}}.$$

To compute the proximal map, we first observe that if θ_1 is known, then by Cai et al. [2010, Theorem 2.1], it follows that

$$\text{prox}^\lambda_{g_2}(x) = \arg\min_z \{e^{\theta_1}\|z\|_{\text{tr}} + \frac{1}{2\lambda}\|x - z\|_F^2\} = S_{e^{\theta_1}\lambda}(x) := U\Sigma_{e^{\theta_1}\lambda}V^T,$$

where $U\Sigma V^T$ is a singular value decomposition, and Σ_β is diagonal with entries $(\Sigma_\beta)_{ii} = \max\{\Sigma_{ii} - \beta, 0\}$. Based on this, we calculate

$$\text{prox}^\lambda_{g_2}(\theta, x) = \arg\min_{(\alpha,z)} \{e^\alpha\|z\|_{\text{tr}} + \frac{1}{2\lambda}\big(\|\theta_1 - \alpha\|^2 + \|x - z\|_F^2\big)\},$$

where $\|\cdot\|$ denotes the Frobenius norm. The minimisers (α, z) satisfy the following system of equations

$$\alpha = \theta_1 + \lambda e^\alpha\|S_{e^\alpha\lambda}(x)\|_{\text{tr}} \Longrightarrow (\alpha - \theta)e^{\theta_1 - \alpha} = \lambda e^{\theta_1}\|S_{e^\alpha\lambda}(x)\|_{\text{tr}}, \tag{54}$$

$$z = S_{\lambda e^\alpha}(x). \tag{55}$$

Solving this system is complicated due to the dependence between α and z and using an iterative solver can be computationally burdensome. Therefore, we have decided to approximate (54) by

$$(\alpha - \theta_1)e^{\theta_1 - \alpha} \approx \lambda e^{\theta_1}\|S_{e^\theta\lambda}(x)\|_{\text{tr}} \Longrightarrow \alpha \approx \theta_1 + W(\lambda e^{\theta_1}\|S_{e^{\theta_1}\lambda}(X^l)\|_{\text{tr}}),$$

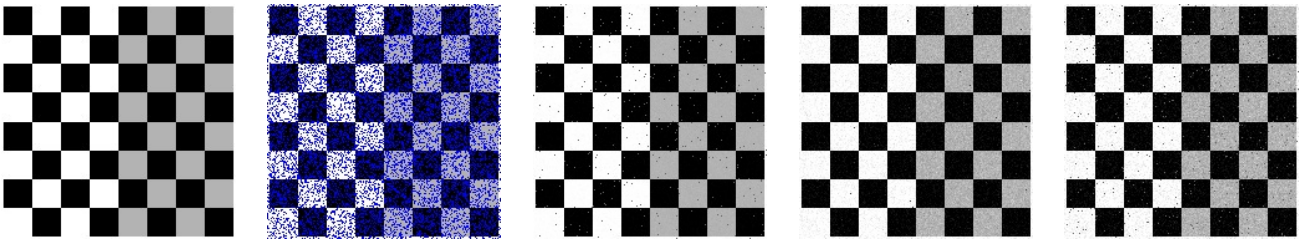
where W is the Lambert W function. Substituting this value of α into (55), we obtain

$$z \approx S_{e^\alpha\lambda}(x).$$

Implementation. To stabilise the implementation of the algorithms, we divide the gradient and proximal mapping terms in the updates of θ_1 and θ_2 by the dimension of the the matrix x, d_x, and the number of observed entries in y, d_y, respectively. We then set $\gamma = 0.01$, and $\lambda = 0.25$ for MYPGD and MYIPLA and $\lambda = 0.01$ for PIPGLA. The pixels of the initial particles are drawn from a normal distribution with mean $\mu = 50$ and scale parameter 10, while the initial values of the parameters θ_1 and θ_2 are drawn from uniform distributions over $[-15, 5]$ and $[-10, 10]$, respectively.

Performance metrics. To asses the performance of our algorithms for low-rank matrix completion, we analyse the normalised mean squared error (NMSE) for both the entire matrix and the missing entries.

Results. Figure 13 displays the original and observed matrices alongside the reconstructed matrices obtained using our different proximal algorithms. The methods are run for 3000 iterations (with a burn-in of 100 iterations) and $N = 10$ particles. The NMSEs for the entire matrix and the missing entries for the final particle cloud are displayed in Table 11, together with the computation times.



(a) Original (b) Observed (c) MYPGD (d) MYIPLA (e) PIPGLA

Figure 13: Low-rank matrix completion. All the algorithms use $N = 10$ particles and are run for 3000 iterations. The blue pixels in (b) represent the mask.

Table 11: Low-rank matrix completion. Normalised mean squared errors (NMSE) for the entire matrix and the missing entries achieved using the final particle cloud with $N = 10$ and 3000 iterations.

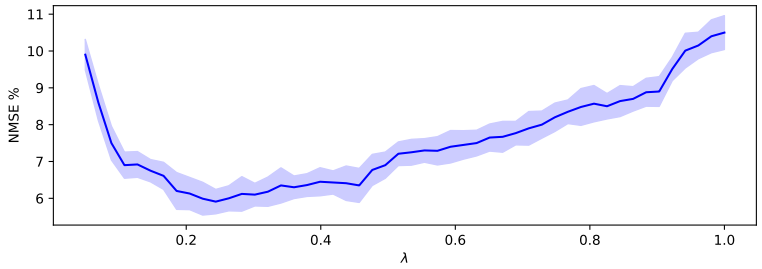
Algorithm	NMSE entire (%)	NMSE missing (%)	Times (min)
MYPGD	1.21 ± 0.49	1.67 ± 0.52	$\mathbf{4.1}$
MYIPLA	$\mathbf{1.13 \pm 0.48}$	$\mathbf{1.53 \pm 0.44}$	4.7
PIPGLA	2.02 ± 0.29	2.11 ± 0.31	5.5

E.6 ABLATION STUDY

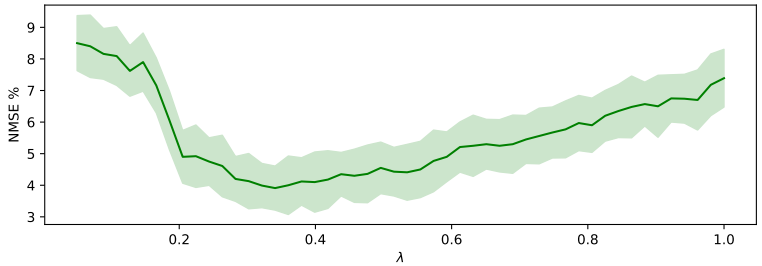
In this section, we analyse how the choice of the regularisation parameter λ in the Moreau–Yosida approximation affects the performance and stability of the algorithm.

Choosing an appropriate value for λ is a challenging task as this parameter controls both the level of regularisation and the closeness to the target, and is closely tied to the step size parameter γ. Durmus et al. [2018] provides some empirical guidance on the choice of γ, λ for sampling tasks. Crucinio et al. [2025] shows that $\lambda \leq \gamma$ generally leads to better results in the case of grad Lipschitz potentials, while one should choose $\lambda \geq \gamma$ for light tail distributions. Adaptive strategies to choose λ have been considered in the optimisation literature (see Oikonomidis et al. [2024] and references therein) but equivalent results for sampling have not been obtained yet.

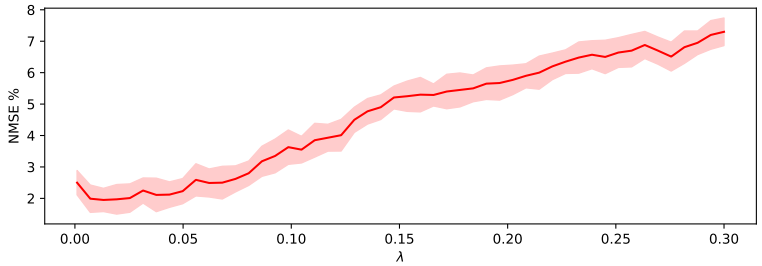
We conduct additional experiments to analyse the impact of the regularisation parameter λ in the Bayesian logistic regression task with Laplace prior. In Figure 14, we report the performance (measured by NMSE) of MYIPLA, MYPGD and PIPGLA algorithms using approximate proximity maps, evaluated over a fine grid of λ values. The step size parameters used are those listed in Table 5: $\gamma = 0.05$ for MYIPLA and MYPGD, and $\gamma = 0.01$ for PIPGLA. Each configuration is run with 100 different random seeds to compute confidence intervals. We observe that our algorithms exhibit stable performance across a broad range of λ values.



(a) MYPGD, $\gamma = 0.05$



(b) MYIPLA, $\gamma = 0.05$



(c) PIPGLA, $\gamma = 0.01$

Figure 14: Normalised MSE (%) for different values of the regularisation parameter λ and a fixed step size γ. Each configuration is run with 100 random seeds for 50 particles and 5000 steps. The proximal map for all algorithms is computed approximately.