
Tuning-Free Maximum Likelihood Training of Latent Variable Models via Coin Betting

Louis Sharrock*

Daniel Dodd*

Christopher Nemeth

Department of Mathematics and Statistics, Lancaster University, UK

Abstract

We introduce two new particle-based algorithms for learning latent variable models via marginal maximum likelihood estimation, including one which is entirely tuning-free. Our methods are based on the perspective of marginal maximum likelihood estimation as an optimization problem: namely, as the minimization of a free energy functional. One way to solve this problem is via the discretization of a gradient flow associated with the free energy. We study one such approach, which resembles an extension of Stein variational gradient descent, establishing a descent lemma which guarantees that the free energy decreases at each iteration. This method, and any other obtained as the discretization of the gradient flow, necessarily depends on a learning rate which must be carefully tuned by the practitioner in order to ensure convergence at a suitable rate. With this in mind, we also propose another algorithm for optimizing the free energy which is entirely learning rate free, based on coin betting techniques from convex optimization. We validate the performance of our algorithms across several numerical experiments, including several high-dimensional settings. Our results are competitive with existing particle-based methods, without the need for any hyperparameter tuning.

1 INTRODUCTION

In statistics and machine learning, probabilistic latent variable models $p_\theta(z, x)$ comprising model *parameters* $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$, unobserved *latent variables* $z \in \mathcal{Z} \subseteq \mathbb{R}^{d_z}$,

and *observations* $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, are widely used to capture the hidden structure of complex data such as images (Bishop, 2006), audio (Smaragdis et al., 2006), text (Blei et al., 2003), and graphs (Hoff et al., 2002). In this paper, we consider the task of estimating the parameters in such models by maximizing the marginal likelihood of the observed data,

$$\theta_* = \arg \max_{\theta \in \Theta} p_\theta(x) := \arg \max_{\theta \in \Theta} \int_{\mathcal{Z}} p_\theta(z, x) dz, \quad (1)$$

and quantifying the uncertainty in the latent variables through the corresponding posterior $p_{\theta_*}(z|x) = p_{\theta_*}(z, x)/p_{\theta_*}(x)$. This framework, which represents a pragmatic compromise between frequentist and Bayesian approaches, is known as the empirical Bayes paradigm (Robbins, 1956; Casella, 1985).

A classical approach for solving the marginal maximum likelihood estimation (MMLE) problem in (1) is the Expectation Maximization (EM) algorithm (Dempster et al., 1977). This iterative method consists of two steps: an expectation step (E-step) and a maximization step (M-step). In the t^{th} iteration, the E-step involves computing the expectation of the log-likelihood with respect to the current posterior distribution $\mu_t := p_{\theta_t}(\cdot|x)$ of the latent variables, viz,

$$Q_t(\theta) = \int_{\mathcal{Z}} \log \pi_\theta(z) \mu_t(z) dz, \quad (E) \quad (2)$$

where $\pi_\theta(z) := p_\theta(z, x)$ denotes the joint density of z and x , given fixed $x \in \mathbb{R}^{d_x}$. Meanwhile, the M-step involves optimizing this quantity with respect to the parameters, namely

$$\theta_{t+1} := \arg \max_{\theta \in \Theta} Q_t(\theta). \quad (M) \quad (3)$$

Under fairly general conditions, this procedure guarantees convergence of the parameters θ_t to a stationary point θ_* of the marginal likelihood, and convergence of the corresponding posterior approximations $p_{\theta_t}(\cdot|x)$ to $p_{\theta_*}(\cdot|x)$ (Wu, 1983; Redner and Walker, 1984; McLachlan and Krishnan, 2007; Balakrishnan et al., 2017). In many applications of interest, neither the E-step nor

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s). *Equal contribution.

the M-step is analytically tractable, in which case it is necessary to use approximations. In particular, the M-step can be approximated using numerical optimization routines (Meng and Rubin, 1993; Liu and Rubin, 1994; Lange, 1995). Meanwhile, the E-step can be approximated via Monte-Carlo methods, leading to the so-called Monte Carlo EM (MCEM) algorithm (Wei and Tanner, 1990; Tanner, 1993; Chan and Ledolter, 1995; Booth and Hobert, 1999; Cappé et al., 1999; Sherman et al., 1999), or otherwise a Robbins-Monro type stochastic approximation procedure (Robbins and Monro, 1951; Kushner and Clark, 1978), resulting in the stochastic approximation EM (SAEM) algorithm (Delyon et al., 1999). In practice, it is often not possible to sample exactly from the current posterior distribution $p_{\theta_t}(\cdot|x)$. In this setting, it is standard to use a Markov chain Monte Carlo (MCMC) approximation for the E-step (e.g., McCulloch, 1997; Levine and Casella, 2001; Fort and Moulines, 2003; Levine and Fan, 2004; Caffo et al., 2005; Fort et al., 2011; Atchadé et al., 2017; Qiu and Wang, 2020; Nijkamp et al., 2020; De Bortoli et al., 2021).

In this paper, we follow a different approach, based on an observation first made in Neal and Hinton (1998), and recently revisited in Kuntz et al. (2023), that the EM algorithm can be viewed as coordinate-descent on a free-energy functional \mathcal{F} (see Sec. 2.1). Leveraging this perspective, Kuntz et al. (2023) obtain a set of particle-based algorithms for optimizing the free energy, and thus for solving (1). A similar approach has since also been studied by Akyildiz et al. (2023). We provide a more detailed discussion of these works in Section 3.

Both in theory and in practice, the performance of these methods relies on a careful choice of hyperparameters such as the learning rate. In particular, the learning rate must be small enough to ensure stability of the parameter updates, whilst also large enough to guarantee convergence at a reasonable rate (Akyildiz et al., 2023, Theorem 1). The situation is additionally complicated by the interdependence between the parameters and the latent variables, as well as the sensitivity of the learning rate to other hyperparameters such as the number of particles. As a result, it is typically necessary to make use of heuristics such as Adagrad (Duchi et al., 2011), Adam (Kingma and Ba, 2015), or RMSProp (Tieleman and Hinton, 2012), or otherwise to resort to computationally expensive quasi-Newton updates (Kuntz et al., 2023, App. C) which exploit a second order approximation of the log-likelihood to better capture its local geometry. In this context, it is natural to ask whether we can obtain alternative algorithms which are more robust to different specifications of the learning rate, or even remove the dependence on the learning rate entirely.

Our contributions. In this paper, we propose two new particle-based algorithms for MMLE in latent variable models, including one which is completely tuning free. Our methods can be applied to a very broad class of latent variable models, namely, any for which the density $p_{\theta}(z, x)$ is differentiable in θ and z . Inspired by Kuntz et al. (2023), both of our algorithms are rooted in the viewpoint of MMLE as the minimization of the free energy functional. Our first approach, SVGD EM, is motivated by recent developments in the theory and application of gradient flows on the space of probability measures (e.g., Otto, 2001; Ambrosio et al., 2008; Duncan et al., 2023). In particular, SVGD EM corresponds to the discretization of a particular gradient flow of the free energy \mathcal{F} on $\Theta \times \mathcal{P}_2(\mathcal{Z})$, and resembles a generalization of the popular Stein variational gradient descent (SVGD) algorithm. Meanwhile, our second approach, Coin EM, is inspired by the parameter-free stochastic optimization methods developed by Orabona and coworkers (Orabona and Pal, 2016a; Orabona and Tommasi, 2017), and their recent extension to the space of probability measures (Sharrock and Nemeth, 2023). In particular, Coin EM leverages a reduction of the minimization of the free energy to two interacting coin betting games. Unlike our first algorithm, or the recent schemes in Akyildiz et al. (2023); Kuntz et al. (2023), Coin EM does not correspond to the time-discretization of any gradient flow, and has no learning rates. It thus bears little resemblance to existing particle-based methods for training latent variable models.

After introducing our algorithms, we study the convergence properties of SVGD EM, establishing a descent lemma which guarantees that this algorithm decreases the free energy at each iteration. We then illustrate the performance of our algorithms on a wide range of examples, including a toy hierarchical model, two Bayesian logistic regression models, two Bayesian neural networks, and a latent space network model. Our results indicate that SVGD EM and Coin EM achieve comparable or superior performance to existing particle-based EM algorithms, with Coin EM removing entirely the need to tune any hyperparameters.

2 MAXIMUM LIKELIHOOD TRAINING OF LATENT VARIABLE MODELS

Notation. We will make use of the following notation. Let $\mathcal{P}_2(\mathcal{Z})$ denote the set of probability measures with finite second moment. Given $\mu \in \mathcal{P}_2(\mathcal{Z})$, let $L^2(\mu)$ denote the space of functions $f : \mathcal{Z} \rightarrow \mathbb{R}$ such that $\int \|f\|^2 d\mu < \infty$, with $\|\cdot\|_{L^2(\mu)}$ and $\langle \cdot, \cdot \rangle_{L^2(\mu)}$ the norm and inner product in this space. For $\mu \in \mathcal{P}_2(\mathcal{Z})$ and measurable $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$, we write $T_{\#}\mu$ for the push-

forward of μ under T . For $\mu, \nu \in \mathcal{P}_2(\mathcal{Z})$, the quadratic Wasserstein distance between μ and ν is defined by $W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathcal{Z} \times \mathcal{Z}} \|z_1 - z_2\|^2 \gamma(dz_1, dz_2)$, where $\Gamma(\mu, \nu)$ is the set of couplings between μ and ν . This is indeed a distance over $\mathcal{P}_2(\mathcal{X})$ (Ambrosio et al., 2008, Chapter 7.1). Finally, we refer to $(\mathcal{P}_2(\mathcal{Z}), W_2)$ as the Wasserstein space.

We will later also write $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ for a positive semi-definite kernel, \mathcal{H}_k for the reproducing kernel Hilbert space (RKHS) associated with this kernel, and $\mathcal{H} := \mathcal{H}_k^{d_z}$ for the product RKHS with elements $f = (f_1, \dots, f_{d_z})$, with $f_i \in \mathcal{H}_k$. We also write $S_\mu : L^2(\mu) \rightarrow \mathcal{H}$ for the integral operator associated with k and the measure μ , defined as $S_\mu f(z) = \int_{\mathbb{R}^d} k(z, w) f(w) \mu(dw)$, and $P_\mu : L^2(\mu) \rightarrow L^2(\mu)$ for the operator $P_\mu = \iota S_\mu$, where $\iota : \mathcal{H} \rightarrow L^2(\mu)$ is the inclusion map, with adjoint $\iota^* = S_\mu$. This map differs from S_μ only in its range.

2.1 The Free Energy

Our approaches will leverage the viewpoint of marginal maximum likelihood estimation as minimization of the free-energy functional $\mathcal{F} : \Theta \times \mathcal{P}(\mathcal{Z}) \rightarrow \mathbb{R}$, defined according to

$$\mathcal{F}(\theta, \mu) := \int \log \frac{\mu(z)}{\pi_\theta(z)} \mu(z) dz. \quad (2)$$

To be precise, we build on the observation made in Neal and Hinton (1998), and recently revisited in Kuntz et al. (2023), that finding $\theta_* = \arg \max_{\theta \in \Theta} p_\theta(x)$ and computing the corresponding posterior $\mu_* = p_{\theta_*}(\cdot|x)$ is equivalent to solving the joint minimization problem

$$(\theta_*, \mu_*) = \arg \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu). \quad (3)$$

In particular, as noted in Neal and Hinton (1998), the EM algorithm corresponds precisely to a coordinate descent scheme applied to \mathcal{F} : given an initial $\theta_0 \in \Theta$, solve

$$\mu_t := \arg \min_{\mu \in \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta_t, \mu), \quad (\text{E})$$

$$\theta_{t+1} := \arg \min_{\theta \in \Theta} \mathcal{F}(\theta, \mu_t), \quad (\text{M})$$

until convergence. Despite the simplicity of this approach, its practical use is limited for more complex models. In particular, it can only be applied when it is possible to solve the two optimization subroutines (i.e., compute the E-step and the M-step) exactly.

2.2 Minimizing the Free Energy using a Stein-Variational Gradient Flow

As pointed out in Kuntz et al. (2023), instead of using (E) and (M) in order to solve (2), a natural alternative

is to use a discretization of a gradient flow associated with (2). We are then faced with several questions. First, what is an appropriate notion of the gradient flow of the functional $\mathcal{F}(\theta, \mu)$? Second, how should we discretize this flow? Regarding the first question, a natural way in which to construct a gradient flow for $\mathcal{F}(\theta, \mu)$ is to consider a Euclidean gradient flow w.r.t. the first argument, and a Wasserstein gradient flow w.r.t. the second argument (e.g., Ambrosio et al., 2008, Chapter 11). In particular, we will say that $(\theta, \mu) : [0, \infty) \rightarrow \Theta \times \mathcal{P}_2(\mathcal{Z})$ is a solution of a ‘Euclidean-Wasserstein’ gradient flow of \mathcal{F} if,

$$\partial_t \theta_t = -\nabla_\theta \mathcal{F}(\theta_t, \mu_t) \quad (4)$$

$$\partial_t \mu_t = -\nabla_\mu \mathcal{F}(\theta_t, \mu_t), \quad (5)$$

where $\nabla_\theta \mathcal{F}(\theta, \mu)$ is the Euclidean gradient of $\mathcal{F}(\cdot, \mu)$ at θ , and $\nabla_\mu \mathcal{F}(\theta, \mu) = -\nabla \cdot (\mu \nabla_{W_2} \mathcal{F}(\theta, \mu))$, with $\nabla_{W_2} \mathcal{F}(\mu, \theta)$ denoting the Wasserstein gradient of $\mathcal{F}(\theta, \cdot)$ at μ , which exists and is given by $\nabla_{W_2} \mathcal{F}(\mu, \theta) = \nabla_z \log(\frac{\mu}{\pi_\theta})$ under mild regularity conditions on $\mu \in \mathcal{P}_2(\mathcal{Z})$ (Ambrosio et al., 2008, Lemma 10.4.13). Explicitly, the gradients in (4) - (5) are given by

$$\nabla_\theta \mathcal{F}(\theta, \mu) = -\int \nabla_\theta \log \pi_\theta(z) \mu(z) dz, \quad (6)$$

$$\nabla_\mu \mathcal{F}(\theta, \mu) = -\nabla \cdot \left(\mu \nabla_z \log \left(\frac{\mu}{\pi_\theta} \right) \right). \quad (7)$$

To obtain an implementable discrete-time algorithm, an obvious choice is to consider an explicit Euler discretization of (4) - (5) which, for $t \in \mathbb{N}_0$, corresponds to¹

$$\theta_{t+1} = \theta_t + \gamma \int \nabla_\theta \log \pi_{\theta_t}(z) \mu_t(z) dz, \quad (8)$$

$$\mu_{t+1} = \left(\text{id} - \gamma \nabla_z \log \left(\frac{\mu_t}{\pi_{\theta_{t+1}}} \right) \right)_\# \mu_t, \quad (9)$$

where $\gamma > 0$ is a step size or learning rate, and id is the identity map. Unfortunately, implementing this scheme would require estimating the density of μ_t based on samples, which is rather challenging.

Inspired by Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016), suppose that we replace the Wasserstein gradient $\nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$ by its image $P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$, under the integral operator P_{μ_t} . This essentially plays the role of the Wasserstein gradient in the RKHS \mathcal{H}_k . Under the assumption that $\lim_{\|z\| \rightarrow \infty} k(z, \cdot) \pi(z) = 0$, we have, using integration by parts (e.g., Liu and Wang, 2016), that

$$P_\mu \nabla_z \log \left(\frac{\mu}{\pi_\theta} \right) (\cdot) \quad (10)$$

$$= - \int [\nabla_z \log \pi_\theta(z) k(z, \cdot) + \nabla_z k(z, \cdot)] \mu(z) dz.$$

¹In a slight abuse of notation, we use t to index both time in the continuous-time dynamics, and the iteration in the discrete-time algorithm. The appropriate meaning should always be clear from context.

Algorithm 1 SVGD EM

- 1: **Input:** number of iterations T , number of particles N , initial particles $\{z_0^i\}_{i=1}^N \sim \mu_0$, initial θ_0 , target π , kernel k , learning rate $\gamma > 0$.
- 2: **for** $t = 0, 1, \dots, T-1$ **do**
- 3:

$$\theta_{t+1} = \theta_t + \frac{\gamma}{N} \sum_{j=1}^N \nabla_{\theta} \log \pi_{\theta_t}(z_t^j)$$

$$z_{t+1}^i = z_t^i + \frac{\gamma}{N} \sum_{j=1}^N \left[k(z_t^j, z_t^i) \nabla_z \log \pi_{\theta_{t+1}}(z_t^j) + \nabla_{z_t^j} k(z_t^j, z_t^i) \right], \quad i \in [N].$$
- 4: **return** θ_T and $\{z_T^i\}_{i=1}^N$.

In this case, the update equations in (8) - (9) become

$$\theta_{t+1} = \theta_t + \gamma \int \nabla_{\theta} \log \pi_{\theta_t}(z) d\mu_t(z), \quad (11)$$

$$\mu_{t+1} = (\text{id} + \gamma \int [\nabla_z \log \pi_{\theta_{t+1}}(z) k(z, \cdot) + \nabla_z k(z, \cdot)] \mu_t(z) dz)_{\#} \mu_t. \quad (12)$$

Finally, we can approximate the two intractable integrals using a set of N interacting particles, $\{z_t^i\}_{i=1}^N$. Based on this approximation, we arrive at Alg. 1.

Naturally, we would like to establish the convergence of this algorithm to the minimizer of $\mathcal{F}(\theta, \mu)$. In App. A.1, we establish that $\mathcal{F}(\theta_t, \mu_t)$ converges exponentially fast along the continuous-time SVGD EM dynamics, under a fairly natural gradient dominance condition. Here, we focus on the discrete-time case. We will require the following assumptions, which extend those introduced in Salim et al. (2022).

Assumption 1. *There exists $B > 0$ such that, for all $z \in \mathcal{Z}$, $\|k(z, \cdot)\|_{\mathcal{H}_k}, \|\nabla_z k(z, \cdot)\|_{\mathcal{H}} \leq B$.*

Assumption 2. *The Hessian H_V of $-\log \pi$ is well defined, and $\exists M > 0$ such that $\|H_V\|_{\text{op}} \leq M$.*

Assumption 3. *For all $\theta \in \Theta$, there exists $\lambda > 0$ such that $p_{\theta}(\cdot|x)$ satisfies Talagrand's T1 inequality with constant λ . That is, for all $\mu \in \mathcal{P}_1(\mathcal{Z})$, $W_1(\mu, p_{\theta}(\cdot|x)) \leq \sqrt{\frac{2\text{KL}(\mu||p_{\theta}(\cdot|x))}{\lambda}}$.*

Remark 1. *These assumptions are rather mild. For example, the T1 inequality holds if $p_{\theta}(\cdot|x)$ satisfies a logarithmic Sobolev inequality with constant $\lambda > 0$ (Vilani, 2008, Theorem 22.17). In turn, this holds if $z \mapsto -\log \pi_{\theta}(z)$ is strongly convex.*

Theorem 1. *Assume that Assumptions 1 - 3 hold. Suppose $0 < \gamma \leq \gamma_*$, where*

$$\gamma_* = \frac{\alpha - 1}{\alpha B^2} \inf_{t \in \mathbb{N}} \left[1 + \|\nabla_z \log p_{\theta_t}(0|x)\| + M \int_{\mathcal{Z}} \|z\| p_{\theta_t}(z|x) dz + M \sqrt{\frac{2\text{KL}(\mu_0||p_{\theta_t}(\cdot|x))}{\lambda}} \right]^{-1} \quad (13)$$

Then, for all $t \geq 0$, there exist positive constants $A_1, A_2 > 0$, given in Appendix (B.2), such that

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_t, \mu_t) \leq -\gamma [A_1 \|\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_{\theta}}}^2 + A_2 \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}}^2]. \quad (14)$$

We prove this Theorem in App. B.2. An immediate consequence is the convergence of $\|\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_{\theta}}}^2$ and $\|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}}^2$ to zero, under appropriate conditions on the learning rate γ . We provide a precise statement of this corollary in App. A.

Naturally, the convergence of this algorithm depends on the learning rate $\gamma > 0$. Indeed, this is the case for any algorithm obtained as the discretization of a gradient flow associated with the free energy functional, including those recently studied in Akyildiz et al. (2023); Kuntz et al. (2023). If γ is too large, the algorithm is unstable and the parameter θ and particles $\{z^i\}_{i=1}^N$ will diverge. If γ is too small, the algorithm will converge slowly. Empirically, Kuntz et al. (2023) noted the significant challenges associated with appropriately setting the learning rate parameter simultaneously for the parameters and the latent variables. This is also observed in our own empirical results (see Sec. 4). To address this issue, we now propose a tuning-free algorithm for minimizing the free energy, which removes entirely the need for user-chosen learning rates, and leads empirically to significantly faster convergence rates.

2.3 Minimizing the Free Energy using Coin Betting

Our approach is based on the learning rate free optimization techniques introduced in Orabona and Pal (2016a,b); Orabona and Tommasi (2017), and their recent extension to optimization problems on spaces of probability measures (Sharrock and Nemeth, 2023). In particular, our method can be viewed as a hybrid between the coin betting techniques from Orabona and Pal (2016a), which we use for the optimization over Θ , and one of the coin sampling algorithms from Sharrock and Nemeth (2023), which we use for the optimization over $\mathcal{P}_2(\mathcal{Z})$. The resulting algorithm, which we term Coin EM, represents an automatic, general-purpose EM algorithm, which does not require a user-defined learning rate, and outperforms existing algorithms across a range of applications (see Sec. 4).

Algorithm 2 Coin EM

- 1: **Input:** number of iterations T , number of particles N , initial particles $\{z_0^i\}_{i=1}^N \sim \mu_0$, initial θ_0 , target π , kernel k .
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3:

$$\theta_t = \theta_0 + \frac{\sum_{s=1}^{t-1} \frac{1}{N} \sum_{j=1}^N \nabla_{\theta} \log \pi_{\theta_s}(z_s^j)}{t} \left(1 + \sum_{s=1}^{t-1} \left\langle \frac{1}{N} \sum_{j=1}^N \nabla_{\theta} \log \pi_{\theta_s}(z_s^j), \theta_s - \theta_0 \right\rangle \right)$$

$$z_t^i = z_0^i + \frac{\sum_{s=1}^{t-1} \frac{1}{N} \sum_{j=1}^N k(z_s^j, z_s^i) \nabla_z \log \pi_{\theta_s}(z_s^j) + \nabla_{z_s^j} k(z_s^j, z_s^i)}{t}$$

$$\times \left(1 + \sum_{s=1}^{t-1} \left\langle \frac{1}{N} \sum_{j=1}^N k(z_s^j, z_s^i) \nabla_z \log \pi_{\theta_s}(z_s^j) + \nabla_{z_s^j} k(z_s^j, z_s^i), z_s^i - z_0^i \right\rangle \right), \quad i \in [N].$$
 - 4: **return** θ_T and $\{z_T^i\}_{i=1}^N$.
-

We begin by recalling the coin betting framework from [Orabona and Pal \(2016a\)](#). Consider a gambler making repeated bets on the outcomes of a series of adversarial coin flips. The gambler’s goal is to maximize their wealth, starting from some initial wealth, $w_0 = 1$. In each round of the game, $t \in \mathbb{N}$, the gambler bets on the outcome of a coin flip, without borrowing any additional money. The gambler’s bet is encoded by $\theta_t \in \mathbb{R}$, where the sign indicates whether the bet is a heads or tails and the absolute value indicates the size of the bet. The gambler’s wealth thus accumulates according to $w_t = w_0 + \sum_{s=1}^t c_s \theta_s$, where $c_t \in \{-1, 1\}$ denotes the outcome of the coin flip. The gambler’s bets are restricted to satisfy $\theta_t = \beta_t w_{t-1}$, where $\beta_t \in [-1, 1]$ denotes a betting fraction, meaning the gambler can only bet a fraction β_t of their accumulated wealth up to time t , and cannot borrow any additional money.

One can use this coin betting game to solve convex optimization problems of the form $\theta_* = \arg \min_{\theta \in \mathbb{R}^d} f(\theta)$, for some convex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. In particular, by considering a coin betting game with outcomes $c_t = -\nabla f(\theta_t)$, and replacing scalar multiplications with scalar products in the framework described above, [Orabona and Pal \(2016a\)](#) proved that, under certain assumptions on the betting strategy, $f(\frac{1}{t} \sum_{s=1}^t \theta_s) \rightarrow f(\theta_*)$ at a rate determined by this strategy. In the case that $|c_t| \leq 1$,² a standard choice for the betting fraction is $\beta_t = \frac{1}{t} \sum_{s=1}^{t-1} c_s$, known as the Krichevsky-Trofimov (KT) betting strategy after [Krichevsky and Trofimov \(1981\)](#). This choice results in the sequence of bets

$$\theta_t = \beta_t w_{t-1} = \frac{\sum_{s=1}^{t-1} c_s}{t} \left(1 + \sum_{s=1}^{t-1} \langle c_s, \theta_s \rangle \right). \quad (15)$$

A similar approach can also be used to solve sampling problems, based on the perspective of sampling as

² If, instead, $|c_t| \leq L$, for some constant $L > 0$, then one can replace c_t by its normalized version. If this constant is unknown, then it can be replaced by an adaptive estimate. We provide details on this approach in App. D.

an optimization problem on the space of probability measures, viz, $\mu_* = \arg \min_{\mu \in \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\mu)$. In this case, several modifications are required. First, in the t^{th} round, one now bets $z_t - z_0$, rather than z_t , where z_0 is a random variable distributed according to some initial $\mu_0 \in \mathcal{P}_2(\mathcal{Z})$. In this case, viewing $z_t : \mathcal{Z} \rightarrow \mathcal{Z}$ as a function that maps $z_0 \mapsto z_t(z_0)$, one can define a series of ‘betting distributions’ $\mu_t \in \mathcal{P}_2(\mathcal{Z})$ as the push-forwards of μ_0 under z_t . This means that, given $z_0 \sim \mu_0$, $z_t \sim \mu_t$.

As shown in [Sharrock and Nemeth \(2023\)](#), this framework can be used to obtain a sampling algorithm by setting $\mathcal{F}(\mu) = \text{KL}(\mu || \pi)$, and considering a betting game in which $c_t = -\mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\mu_t)(z_t)$. The unknown $(\mu_t)_{t \in \mathbb{N}}$ are then approximated using a set of N particles. This results in a learning-rate free analogue of SVGD, referred to as Coin SVGD. Empirically, this approach has demonstrated competitive performance with SVGD, with no need to tune a learning rate ([Sharrock and Nemeth, 2023; Sharrock et al., 2023](#)).

Extending this approach, we can obtain a learning-rate free algorithm for MMLE in latent variable models. In particular, to minimize the variational free energy $\mathcal{F}(\theta, \mu)$, we consider a coin betting game involving two players, whose outcomes in the t^{th} round are given by $c_t^{(1)} = -\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t)$ and $c_t^{(2)} = -\mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$, respectively.³ By combining the original coin betting algorithm in [Orabona and Pal \(2016a,b\)](#) to optimize $\mathcal{F}(\theta, \mu)$ over $\theta \in \Theta$, and the Coin SVGD algorithm in [Sharrock and Nemeth \(2023\)](#) to optimize over $\mu \in \mathcal{P}_2(\mathcal{Z})$, we now have a learning rate free method for solving (2). This approach, which we refer to as Coin EM, is summarised in Alg. 2.

In Theorem 2, we provide a convergence result for a version of Coin EM in the population limit. Analyzing

³ Alternatively, one can consider this as a betting game involving a single gambler, who bets on two sets of interacting outcomes given by $c_t^{(1)} = -\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t)$ and $c_t^{(2)} = -\mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$.

the convergence of this scheme is extremely challenging, and requires an additional technical assumption that it is unclear how to verify in practice. Nonetheless, we include this result in the interest of completeness. We will require the following assumptions.

Assumption 4. *The function $(\theta, z) \mapsto -\log \pi_\theta(z)$ is strictly convex.*

Assumption 5. *There exists $L > 0$ such that $\|\nabla_\theta \log \pi_{\theta_t}(z)\|, \|\nabla_z \log(\frac{\mu_t}{\pi_{\theta_t}})(z_t)\| \leq L$ for $t \in [T]$.*

Theorem 2. *Assume that Assumptions 4, 5, and 8 (App. B.4) hold. Then the marginal likelihood $\theta \mapsto p_\theta(x)$ admits a unique maximizer θ^* . In addition, writing $\mu^* = p_{\theta^*}(\cdot|x)$, it holds for all $T \geq 1$ that*

$$\mathcal{F}\left(\frac{1}{T} \sum_{t=1}^T \theta_t, \frac{1}{T} \sum_{t=1}^T \mu_t\right) - \mathcal{F}(\theta^*, \mu^*) = \tilde{O}\left(\frac{1}{\sqrt{T}}\right). \quad (16)$$

We provide a full statement of this result in App. A.3, and its proof in App. B.4.

Computational Complexity. The computational complexity of both SVGD EM and Coin EM is $\mathcal{O}(N^2 T [\text{eval. cost of } (\nabla_\theta \log \pi, \nabla_z \log \pi)])$. To alleviate the cost w.r.t. the number of particles N , one can approximate the sum $\sum_{j=1}^N$ over the particles by subsampling (Li et al., 2020), or by using a random feature expansion of the kernel (Rahimi and Recht, 2007). Meanwhile, in big data settings where computing $(\nabla_\theta \log \pi, \nabla_z \log \pi)$ is expensive, these gradients can be replaced with stochastic estimates obtained by subsampling mini-batches of the data (e.g., Welling and Teh, 2011; Liu and Wang, 2016).

3 RELATED WORK

Comparison with Kuntz et al. (2023) and Akyildiz et al. (2023). In a recent paper, Kuntz et al. (2023) revisited the perspective of MMLE as the minimization of the free energy, proposing the gradient flow in (4) - (5) to minimize this functional. In contrast to us, their algorithms are based on the observation that (4) - (5) is a Fokker-Planck equation satisfied by the law of

$$d\theta_t = \left[\int \nabla_\theta \log \pi_{\theta_t}(z_t) d\mu_t(z) \right] dt, \quad (17)$$

$$dz_t = \nabla_z \log \pi_{\theta_t}(z_t) dt + \sqrt{2} dw_t. \quad (18)$$

By approximating this SDE using a system of interacting particles $\{z_t^j\}_{j=1}^N$, and discretizing in time, Kuntz et al. (2023) obtain the particle gradient descent (PGD) algorithm. Akyildiz et al. (2023) have since analyzed an extension of this algorithm, which includes a noise term in the θ dynamics in (18), allowing them to obtain a non-asymptotic concentration bound for θ_t .

Kuntz et al. (2023) also develop two variants of PGD. The first is Particle Quasi-Newton, which includes a preconditioning term in the θ dynamics (Kuntz et al., 2023, App. C). In principle, one could obtain a similar version of SVGD EM based on the techniques in Detommaso et al. (2018). The second is Particle Marginal Gradient Descent, in which the θ update in (18) is replaced by $\theta_t = \theta_*(\mu_t^N)$, where $\theta_*(\mu) = \arg \min_{\mu \in \mathcal{P}(\mathcal{Z})} \mathcal{F}(\theta, \mu)$, which can be applied whenever it is possible to compute $\mu \mapsto \theta_*(\mu)$ in closed form (Kuntz et al., 2023, App. D). In a similar vein, we can obtain marginal variants of our algorithms. These are given in App. C.

Learning-rate free methods for optimization and sampling. The idea of using coin betting for parameter free online learning was first introduced in Orabona and Pal (2016a,b) and has since been extensively developed by Orabona and coworkers (Orabona and Tommasi, 2017; Jun et al., 2017; Cutkosky and Orabona, 2018; Jun and Orabona, 2019; Chen et al., 2022a,b). On the other hand, aside from Sharrock and Nemeth (2023); Sharrock et al. (2023), the literature on learning-rate free sampling methods is sparse. In practice, the standard technique when using gradient-based methods is to use a grid search, running the algorithm of choice for multiple learning rates, and selecting the value which minimizes an appropriately chosen metric. While there have been some efforts to automate the design of effective learning rate schedules (Chen et al., 2016; Li et al., 2016; Coullon et al., 2021; Kim et al., 2022), typically these approaches still rely on various hyperparameters.

4 NUMERICAL EXPERIMENTS

We now benchmark SVGD EM (Alg. 1) and Coin EM (Alg. 2) against other comparable approaches. In all experiments, we use the RBF kernel $k(x, x') = \exp(-\frac{1}{h} \|x - x'\|^2)$, with the bandwidth h chosen according to the median heuristic (Liu and Wang, 2016). We provide additional experimental details in App. E, and additional numerical results in App. F. The code to reproduce our results can be found on GitHub at <https://github.com/chris-nemeth/coinem>.

4.1 Toy hierarchical model

We begin by considering a toy hierarchical model from Kuntz et al. (2023). Suppose that we observe data $x = (x_1, \dots, x_{d_z})^\top$, generated according to $x_i | z_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(x_i | z_i, 1)$, where $z_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\theta, 1)$, for some parameter $\theta \in \mathbb{R}$. Our model is thus given by

$$p_\theta(z, x) = \prod_{i=1}^{d_z} \frac{1}{2\pi} \exp \left[-\frac{(z_i - \theta)^2}{2} - \frac{(x_i - z_i)^2}{2} \right]. \quad (19)$$

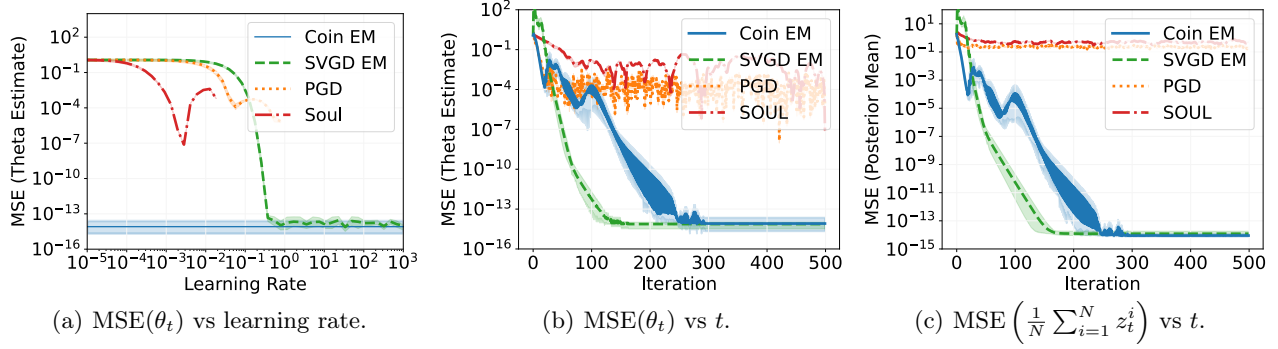


Figure 1: **Results for the toy hierarchical model.** MSE of the parameter estimate θ_t as a function of the learning rate after $T = 500$ iterations (a); and MSE of the parameter estimate (b) and the posterior mean (c) as a function of the number of iterations, using the optimal learning rate from (a).

In this case, the marginal likelihood $\theta \mapsto p_\theta(x)$ has a unique maximum given by $\theta_* = d_z^{-1} \sum_{i=1}^{d_z} x_i$, and one can obtain an explicit expression for the corresponding posterior $p_{\theta_*}(\cdot|x)$ (Kuntz et al., 2023, App. E.1).

In Fig. 1, we evaluate the performance of SVGD EM and Coin EM on this model, setting $d_z = 100$ and $\theta = 1$. We also include results for PGD (Kuntz et al., 2023) and SOUL (De Bortoli et al., 2021). In this case, both of our methods generate parameters θ_t which converge rapidly to θ_* , and particles $\{z_t^i\}_{i=1}^N$ whose mean converges to the corresponding posterior mean (Fig. 1(b) - 1(c)). Even in this toy example, it is clear that PGD, SOUL, and to a lesser extent SVGD EM, are very sensitive to the learning rate (Fig. 1(a)). If the learning rate is too small, then convergence is slow; if it is too large, then the parameter estimates are unstable and may fail to converge (see also Fig. 8 in App. F.1). Coin EM circumvents this problem entirely, obtaining comparable or superior performance to the competing methods, with no need to tune a learning rate.

In Fig. 2, we further investigate the performance of our methods, plotting the posterior variance estimates from Coin EM and SVGD EM in the case $d_z = 1$.

In this case, there is a significant bias when using a small number of particles, an observation which also holds true for other particle-based methods such as PGD (Kuntz et al., 2023, Figure 2). This should not be a surprise: even if the parameters were fixed, the SVGD updates in Alg. 1 would only converge to the true posterior in the mean-field limit (e.g., Liu, 2017; Duncan et al., 2023; Korba et al., 2020; Salim et al., 2022). Nonetheless, as is evident in Fig. 2, this bias can be all but eliminated by taking a sufficiently large number of particles. In any case, in terms of posterior variance, Coin EM and SVGD EM are once again comparable or superior to the optimal performance of PGD and SOUL (Fig. 2(c)). Additional results, illustrating the robustness of our method to changes in the number of particles (Fig. 6, Fig. 7) and the initialization (Fig. 9), can be found in App. F.1.

4.2 Bayesian logistic regression

We next consider a standard Bayesian logistic regression with Gaussian priors, fit using the Wisconsin dataset Wolberg and Mangasarian (1990). In this case, the latent variables are the regression weights. We place an

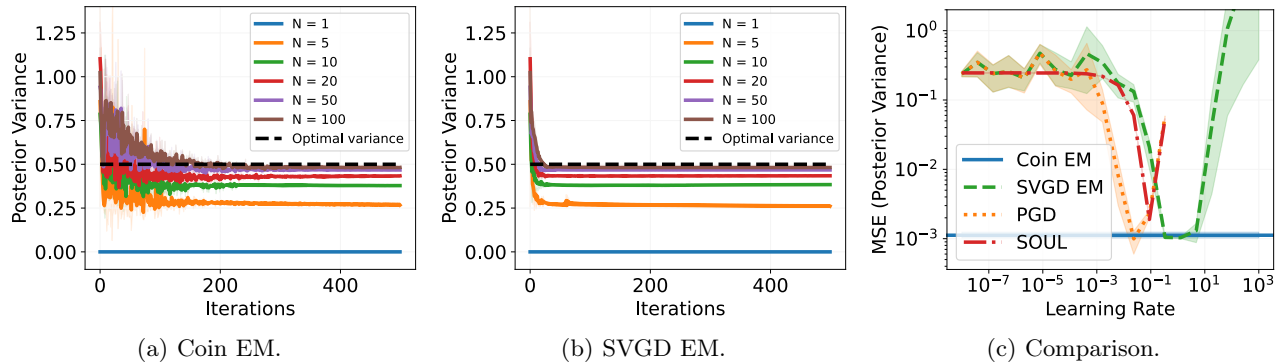


Figure 2: **Additional results for the toy hierarchical model.** Estimates for the posterior variance in the case $d_z = 1$ obtained using (a) Coin EM and (b) SVGD EM, as a function of the number of iterations. In (c), we plot the MSE of the posterior variance estimate as a function of the learning rate, for Coin EM, SVGD EM, PGD, and SOUL, after $T = 250$ iterations and with $N = 50$ particles.

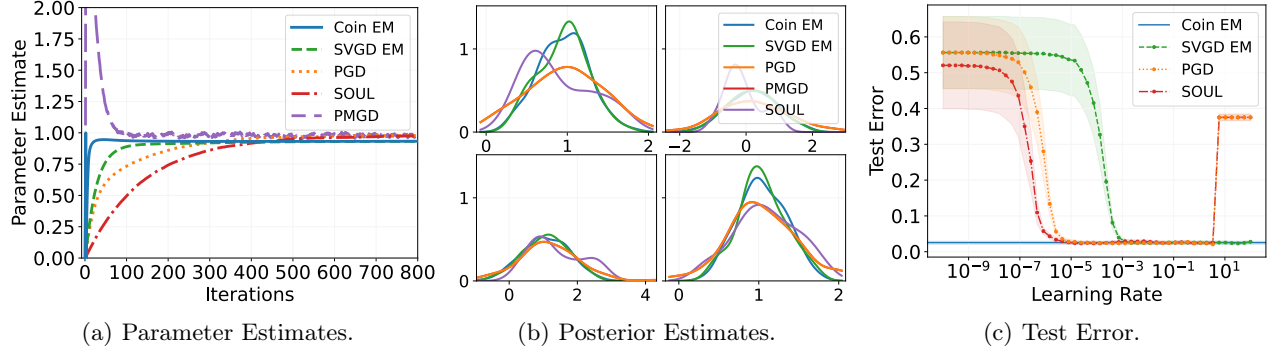


Figure 3: **Results for the Bayesian logistic regression.** Plots of (a) the sequence of parameter estimates θ_t initialized at zero, (b) the kernel density estimate of four components of the posterior approximation $\hat{\mu}_{800}^n = \frac{1}{n} \sum_{j=1}^n \delta_{z_{800}^j}$, (c) the test error as a function of the learning rate.

isotropic Gaussian prior $\mathcal{N}(\theta \mathbf{1}_{d_z}, 51_{d_z})$ on the weights, and aim to estimate the unique maximizer θ_* of the marginal likelihood. We provide full details on this model in App. E.2, and additional results for an alternative model in App. F.3.

In Fig. 3, we compare the performance of our algorithms with PGD (Kuntz et al., 2023), PMGD (Kuntz et al., 2023), and SOUL (De Bortoli et al., 2021). We first plot an illustrative sequence of parameter estimates (Fig. 3(a)) for each method, initialized at zero, using $N = 100$ particles and $T = 800$ iterations. All methods converge to approximately the same θ_* . In this case, Coin EM converges noticeably faster than its competitors. The same is true when the parameter is initialized further from θ_* , in which case the shorter transient period exhibited by Coin EM is even more pronounced (Fig. 10 in App. F.2). Interestingly, by increasing the learning rates of PGD, PMGD, and SOUL, one can improve their convergence rates, but this comes at the cost of a (significant) bias in the resulting parameter estimate (Fig. 11 in App. F.2).

Meanwhile, the posterior estimates obtained by each method are rather similar, as is their predictive performance. In fact, in this case the predictive power of the posterior approximations obtained by all of the methods are robust both to the choice of learning rate (Fig. 3(c)) and the number of particles (see Fig. 12 in App. F.2). This can be attributed to the relatively simple nature of the posterior, which is both peaked and unimodal; see also (Kuntz et al., 2023, Sec. 3.1), (De Bortoli et al., 2021, Sec. 4.1).

4.3 Bayesian neural network

We now consider an example with a notably more complex posterior, namely, a Bayesian neural network (see App. E.4 for full details). We consider the setting described in Section 5 in Yao et al. (2022); Section 3.2 in Kuntz et al. (2023), using a simple two-layer neural

network to classify MNIST images (Lecun et al., 1998). The input layer consists of 40 nodes and 784 inputs, while the output layer has 2 nodes. The latent variables are the weights $w \in \mathbb{R}^{784 \times 40}$ and $v \in \mathbb{R}^{2 \times 40}$ of the output and input layers. We place zero mean isotropic Gaussian priors on the weights, with variances $e^{2\alpha}$ and $e^{2\beta}$, respectively. Rather than assigning hyperpriors to $\theta = (\alpha, \beta)$, we will learn these parameters from data.

In Fig. 4, we plot the test error achieved by Coin EM, SVGD EM, PGD, and SOUL for different numbers of particles.⁴ For the learning-rate dependent methods, we use the best learning-rate as determined by a search over a fine grid (see Fig. 14 in App. F.4). In this example, Coin EM offers comparable or superior performance to the best competing method (PGD'). While the initial convergence of PGD' is typically slightly faster, the asymptotic error of Coin EM is consistently lower, particularly for small numbers of particles. In fact, our methods are generally more robust to changes in the number of particles (Fig. 15 in App. F.4), the need for a scaling heuristic to avoid ill conditioning (Fig. 14 in App. F.4), and, of course, the learning rate (Fig. 16 in App. F.4). In App. F.5, we provide additional results for another Bayesian neural network (see App. E.5 for details), with similar conclusions.

4.4 Latent space network model

Finally, we consider a latent space network model (Hoff et al., 2002; Loyal and Chen, 2023). Such models assume that each node in a network has an unobserved latent position in a low-dimensional Euclidean space, and that the probability of a link between two nodes i and j depends on the distance between their latent variables $\|z_{(i)} - z_{(j)}\|$. Latent space network models can account for transitivity, homophily, and other net-

⁴We also include results for SVGD EM, PGD, and SOUL when using a scaling heuristic recommended in (Kuntz et al., 2023, Sec. 2), designed to stabilize the updates and avoid ill-conditioning (SVGD EM', PGD', and SOUL').

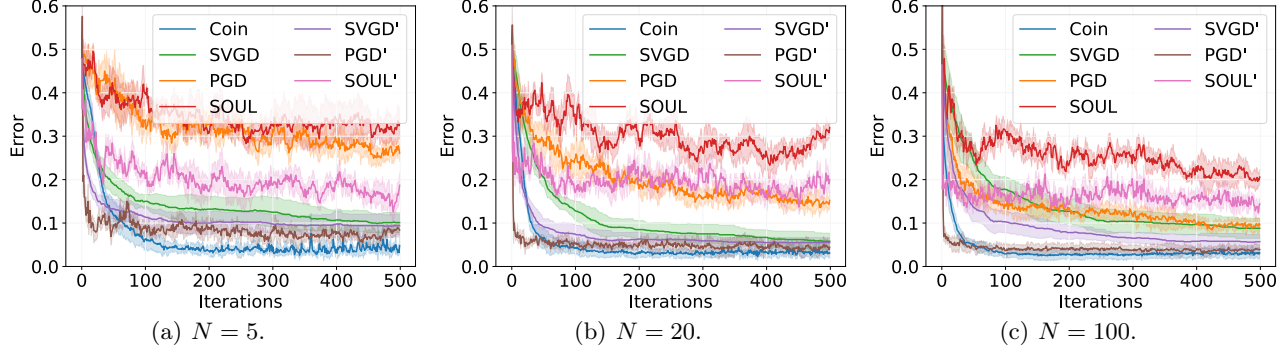


Figure 4: **Results for the Bayesian neural network model.** Test error over $T = 500$ training iterations, for different N . For all learning-rate dependent methods, we use the best learning rate as determined by Fig. 14.

work properties, and provide a natural way to analyze social network data. In particular, the estimated latent positions can be used to visualize the network, and also for downstream tasks such as node classification or community detection through clustering.

Here, we consider fitting a latent space network model to a dataset curated by [Beveridge and Chemers \(2018\)](#), which consists of a sequence of binary undirected networks indicating whether or not an interaction occurred between two characters in the TV series *Game of Thrones*, with one network for each of the eight seasons (see App. E.6 for further details).

In Fig. 5, we plot the latent representation of the character interactions obtained using Coin EM. Plots for the other algorithms are given in App. F.6. In this case, we find that the latent representations obtained via Coin EM successfully capture several natural groupings of the characters, and how these groupings evolve through the first three seasons. For example, the latent representations of characters with significant interactions (e.g., Daenerys and Jorah in Season 1; Arya and the Hound in Season 3) tend to appear in similar clusters. Meanwhile, this is not the case for the latent representations determined by PGD (Fig. 19 in App. F.6) or SOUL (Fig. 20 in App. F.6), both of which fail to capture known relationships between characters, or how these evolve.

5 DISCUSSION

Summary. In this paper, we introduced two new algorithms for MMLE in latent variable models, including one which is entirely tuning-free. Our first algorithm, SVGD EM (Sec. 2.2), can be viewed as a particular form of gradient descent on the free-energy functional \mathcal{F} over the product space $\Theta \times \mathcal{P}_2(\mathcal{Z})$. Our second algorithm, Coin EM (Sec. 2.3), is entirely different, and leverages coin betting ideas to remove any dependence on learning rates.

Limitations and Future Work. We highlight several limitations, as well as some directions for future work. First, similar to SVGD, our algorithms have a $\mathcal{O}(N^2)$ cost per update. This compares unfavourably to the $\mathcal{O}(N)$ cost for PGD ([Kuntz et al., 2023](#)). Second, our convergence results for SVGD were derived in the population limit. In principle, one could obtain finite-particle convergence rates for SVGD EM using the techniques in [Shi and Mackey \(2023\)](#), or by considering ‘virtual particle’ variants of SVGD EM, extending the approach in [Das and Nagaraj \(2023\)](#). Finally, even in the population limit, the convergence result for Coin EM was obtained under a technical sufficient condition, which it is unclear how to verify in practice. The problem of establishing the convergence of Coin EM without this assumption remains an interesting direction for future work.

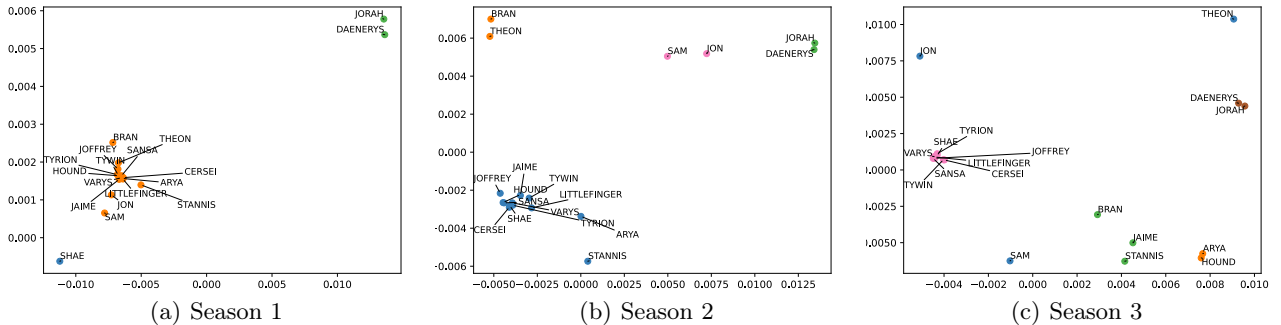


Figure 5: **Results for the latent space network model.** Mean of the particles $\{z_T^i\}_{i=1}^N$ output by Coin EM after $T = 500$ iterations. Each node of the network represents a Game of Thrones character.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. We are grateful to Juan Kuntz for many insightful discussions. LS and CN were supported by the Engineering and Physical Sciences Research Council (EPSRC), grant number EP/V022636/1. CN acknowledges further support from the EPSRC, grant numbers EP/R01860X/1 and EP/Y028783/1. DD was supported by the EPSRC-funded STOR-i Centre for Doctoral Training, grant number EP/S022252/1.

References

- Akyıldız, Ö. D., Crucinio, F. R., Girolami, M., Johnston, T., and Sabanis, S. (2023). Interacting Particle Langevin Algorithm for Maximum Marginal Likelihood Estimation. *arXiv preprint*. 2, 4, 6
- Ambrosio, L., Gigli, N., and Giuseppe Savaré (2008). *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Birkhäuser, Basel. 2, 3, 25
- Atchadé, Y. F., Fort, G., and Moulines, E. (2017). On Perturbed Proximal Gradient Algorithms. *Journal of Machine Learning Research*, 18:1–33. 2
- Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120. 1
- Beveridge, A. and Chemers, M. (2018). The Game of Game of Thrones: Networked Concordances and Fractal Dramaturgy. In *Reading Contemporary Serial Television Universes*, pages 201–225. Routledge. 9, 30
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag, New York. 1
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022. 1
- Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):265–285. 2
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs. 27
- Brenier, Y. (1991). Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417. 17
- Caffo, B. S., Jank, W., and Jones, G. L. (2005). Ascent-based Monte Carlo expectation–maximization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):235–251. 2
- Cappé, O., Doucet, A., Lavielle, M., and Moulines, E. (1999). Simulation-based methods for blind maximum-likelihood filter identification. *Signal Processing*, 73(1):3–25. 2
- Casella, G. (1985). An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 39(2):83–87. 1
- Chan, K. S. and Ledolter, J. (1995). Monte Carlo EM Estimation for Time Series Models Involving Counts. *Journal of the American Statistical Association*, 90(429):242–252. 2
- Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. (2016). Bridging the Gap between Stochastic Gradient MCMC and Stochastic Optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, Cadiz, Spain. 6
- Chen, K., Cutkosky, A., and Orabona, F. (2022a). Implicit Parameter-free Online Learning with Truncated Linear Models. In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory (ALT 2022)*, Paris, France. 6
- Chen, K., Langford, J., and Orabona, F. (2022b). Better Parameter-Free Stochastic Optimization with ODE Updates for Coin-Betting. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, Online. 6
- Coullon, J., South, L., and Nemeth, C. (2021). Efficient and Generalizable Tuning Strategies for Stochastic Gradient MCMC. *arXiv preprint*. 6
- Cutkosky, A. and Orabona, F. (2018). Black-Box Reductions for Parameter-free Online Learning in Banach Spaces. In *Proceedings of the 31st Annual Conference on Learning Theory (COLT 2018)*, Stockholm, Sweden. 6
- Das, A. and Nagaraj, D. (2023). Provably Fast Finite Particle Variants of SVGD via Virtual Particle Stochastic Approximation. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, LA. 9
- De Bortoli, V., Durmus, A., Pereyra, M., and Vidal, A. F. (2021). Efficient stochastic optimisation by unadjusted Langevin Monte Carlo. *Statistics and Computing*, 31(3):29. 2, 7, 8, 28, 32, 33
- Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128. 2

- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38. 1
- Detommaso, G., Cui, T., Spantini, A., Marzouk, Y., and Scheichl, R. (2018). A Stein variational Newton method. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NIPS 2018)*, Montreal, Canada. 6
- Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. Technical report, University of California, Irvine, School of Information and Computer Sciences. 28, 30
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159. 2, 27, 30, 31
- Duncan, A., Nüsken, N., and Szpruch, L. (2023). On the geometry of Stein variational gradient descent. *Journal of Machine Learning Research*, 24:1–40. 2, 7, 15, 16
- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press. 37
- Fort, G. and Moulines, E. (2003). Convergence of the Monte Carlo expectation maximization for curved exponential families. *The Annals of Statistics*, 31(4):1220–1259. 2
- Fort, G., Moulines, E., and Priouret, P. (2011). Convergence of adaptive and interacting Markov chain Monte Carlo algorithms. *The Annals of Statistics*, 39(6):3262–3289. 2
- Gershman, S. J., Hoffman, M. D., and Blei, D. M. (2012). Nonparametric Variational Inference. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, Edinburgh, UK. 28
- Hernandez-Lobato, J. M. and Adams, R. P. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Lille, France. 29
- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098. 1, 8, 30
- Jun, K.-S. and Orabona, F. (2019). Parameter-Free Online Convex Optimization with Sub-Exponential Noise. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT 2019)*, Phoenix, AZ. 6
- Jun, K.-S., Orabona, F., Wright, S., and Willett, R. (2017). Online Learning for Changing Environments using Coin Betting. *Electronic Journal of Statistics*, 11. 6
- Kim, S., Song, Q., and Liang, F. (2022). Stochastic gradient Langevin dynamics with adaptive drifts. *Journal of Statistical Computation and Simulation*, 92(2):318–336. 6
- Kingma, D. P. and Ba, J. (2015). Adam: a method for stochastic optimisation. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, pages 1–13, San Diego, CA. 2
- Korba, A., Salim, A., Arbel, M., Luise, G., and Gretton, A. (2020). A Non-Asymptotic Analysis for Stein Variational Gradient Descent. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada. 7, 15, 16, 18, 20, 21, 22
- Krichevsky, R. E. and Trofimov, V. K. (1981). The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207. 5
- Kuntz, J., Lim, J. N., and Johansen, A. M. (2023). Particle algorithms for maximum likelihood training of latent variable models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, Valencia, Spain. 2, 3, 4, 6, 7, 8, 9, 15, 18, 20, 25, 28, 29, 33, 34, 36
- Kushner, H. J. and Clark, D. S. (1978). *Stochastic approximation methods for constrained and unconstrained systems*. Springer-Verlag, New York. 2
- Lange, K. (1995). A Gradient Algorithm Locally Equivalent to the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):425–437. 2
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 8, 29
- Levine, R. A. and Casella, G. (2001). Implementations of the Monte Carlo EM Algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439. 2
- Levine, R. A. and Fan, J. (2004). An automated (Markov chain) Monte Carlo EM algorithm. *Journal of Statistical Computation and Simulation*, 74(5):349–360. 2
- Li, C., Chen, C., Carlson, D., and Carin, L. (2016). Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, AZ. 6

- Li, L., Li, Y., Liu, J.-G., Liu, Z., and Lu, J. (2020). A stochastic version of Stein variational gradient descent for efficient sampling. *Communications in Applied Mathematics and Computational Science*, 15(1):37–63. [6](#)
- Liu, C. and Rubin, D. B. (1994). The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence. *Biometrika*, 81(4):633–648. [2](#)
- Liu, Q. (2017). Stein variational gradient descent as gradient flow. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, pages 3118–3126, Red Hook, NY. [7](#)
- Liu, Q. and Wang, D. (2016). Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Proceedings of the 30th Conference on Neural Information Processings Systems (NIPS 2016)*, Barcelona, Spain. [3](#), [6](#), [29](#)
- Loyal, J. D. and Chen, Y. (2023). A Bayesian Nonparametric Latent Space Approach to Modeling Evolving Communities in Dynamic Networks. *Bayesian Analysis*, 18(1):49–77. [8](#)
- McCulloch, C. E. (1997). Maximum Likelihood Algorithms for Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 92(437):162–170. [2](#)
- McLachlan, G. J. and Krishnan, T. (2007). *The EM Algorithm and Extensions*. John Wiley and Sons, 2nd edition. [1](#)
- Meng, X.-L. and Rubin, D. B. (1993). Maximum Likelihood Estimation via the ECM Algorithm: A General Framework. *Biometrika*, 80(2):267–278. [2](#)
- Neal, R. M. and Hinton, G. E. (1998). A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 355–368. Springer Netherlands, Dordrecht. [2](#), [3](#)
- Nijkamp, E., Pang, B., Han, T., Zhu, S.-C., and Wu, Y. N. (2020). Learning Multi-layer Latent Variable Model via Variational Optimization of Short Run MCMC for Approximate Inference. In *European Conference on Computer Vision*, pages 361–378, Online. [2](#)
- Orabona, F. and Cutkosky, A. (2020). Tutorial on Parameter-Free Online Learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, Online. [25](#), [31](#)
- Orabona, F. and Pal, D. (2016a). Coin Betting and Parameter-Free Online Learning. In *Proceedings of the 30th Conference on Neural Information Processings Systems (NIPS 2016)*, Barcelona, Spain. [2](#), [4](#), [5](#), [6](#), [25](#)
- Orabona, F. and Pal, D. (2016b). Parameter-Free Convex Learning Through Coin Betting. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016): AutoML Workshop*, New York, NY. [4](#), [5](#), [6](#)
- Orabona, F. and Tommasi, T. (2017). Training Deep Networks without Learning Rates Through Coin Betting. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA. [2](#), [4](#), [6](#), [26](#)
- Otto, F. (2001). The Geometry of Dissipative Evolution Equations: The Porous Medium Equation. *Communications in Partial Differential Equations*, 26(1-2):101–174. [2](#)
- Qiu, Y. and Wang, X. (2020). Stochastic Approximate Gradient Descent via the Langevin Algorithm. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20)*, New York, NY. [2](#)
- Rahimi, A. and Recht, B. (2007). Random Features for Large-Scale Kernel Machines. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007)*, Vancouver, Canada. [6](#)
- Redner, R. A. and Walker, H. F. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239. [1](#)
- Robbins, H. (1956). An empirical Bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 157–164. [1](#)
- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407. [2](#)
- Salim, A., Sun, L., and Peter Richtárik (2022). A Convergence Theory for SVGD in the Population Limit under Talagrand’s Inequality T1. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, Online. [4](#), [7](#), [22](#), [24](#)
- Sharrock, L., Mackey, L., and Nemeth, C. (2023). Learning Rate Free Sampling in Constrained Domains. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, LA. [5](#), [6](#)
- Sharrock, L. and Nemeth, C. (2023). Coin Sampling: Gradient-Based Bayesian Inference without Learning Rates. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, Honolulu, HI. [2](#), [4](#), [5](#), [6](#), [17](#), [25](#), [26](#)
- Sherman, R. P., Ho, Y.-Y. K., and Dalal, S. R. (1999). Conditions for convergence of Monte Carlo EM sequences with an application to product diffusion modeling. *The Econometrics Journal*, 2(2):248–267. [2](#)

- Shi, J. and Mackey, L. (2023). A Finite-Particle Convergence Rate for Stein Variational Gradient Descent. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, LA. [9](#)
- Smaragdis, P., Raj, B., and Shashanka, M. (2006). A Probabilistic Latent Variable Model for Acoustic Modeling. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems: Workshop on Advances in Models for Acoustic Processing (NIPS 2006)*, Vancouver, Canada. [1](#)
- Tanner, M. A. (1993). *Tools for Statistical Inference*. Springer-Verlag, New York, NY, 2nd edition. [2](#)
- Tieleman, T. and Hinton, G. E. (2012). Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31. [2](#)
- Turnbull, K., Lunagomez, S., Nemeth, C., and Airolidi, E. (2019). Latent space modelling of hypergraph data. *arXiv preprint*. [30](#)
- van Erven, T. and Harremos, P. (2014). Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820. [21](#)
- Villani, C. (2003). *Topics in Optimal Transportation*. American Mathematical Society, Providence, Rhode Island. [23](#)
- Villani, C. (2008). *Optimal Transport: Old and New*. Springer-Verlag, Berlin. [4](#)
- Wei, G. C. G. and Tanner, M. A. (1990). 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. [2](#)
- Welling, M. and Teh, Y. W. (2011). Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, Bellevue, WA. [6](#)
- Wolberg, W. H. and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23):9193–9196. [7](#), [28](#)
- Wu, C. (1983). On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103. [1](#)
- Yao, Y., Vehtari, A., and Gelman, A. (2022). Stacking for nonmixing Bayesian computations: The curse and blessing of multimodal posteriors. *Journal of Machine Learning Research*, 23(79):1–45. [8](#), [29](#)

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
 - (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
 - (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]
 - (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
 - (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]

Tuning-Free Maximum Likelihood Training of Latent Variable Models via Coin Betting: Supplementary Materials

A ADDITIONAL THEORETICAL RESULTS

We will require the following additional notation. We will write $\mathcal{U} = (\Theta, \|\cdot\|_{\mathbb{R}^{d_\theta}}) \times (\mathcal{H}_k^{d_z}, \|\cdot\|_{\mathcal{H}_k^{d_z}})$ denote the Cartesian product of the Euclidean space $(\Theta, \|\cdot\|_{\mathbb{R}^{d_\theta}})$ and the Hilbert product space $(\mathcal{H}_k^{d_z}, \|\cdot\|_{\mathcal{H}_k^{d_z}})$, with norm $\|\cdot\|_{\mathcal{U}}$ defined according to $\|(\theta, f)\|_{\mathcal{U}}^2 = \|\theta\|_{\mathbb{R}^{d_\theta}}^2 + \|f\|_{\mathcal{H}_k^{d_z}}^2$ for $\theta \in \Theta$ and $f \in \mathcal{H}_k^{d_z}$. In addition, we will write $\nabla_{\mathcal{U}}\mathcal{F} = (\nabla_{\theta}\mathcal{F}, S_{\mu}\nabla_{W_2}\mathcal{F}) \in \mathcal{U}$.

A.1 SVGD EM: Continuous Time Results

In this section, we study the properties of the SVGD EM gradient flow, which we recall is given by

$$\frac{\partial \theta_t}{\partial t} = -\nabla_{\theta}\mathcal{F}(\theta_t, \mu_t), \quad \nabla_{\theta}\mathcal{F}(\theta_t, \mu_t) = -\int \nabla_{\theta} \log \pi_{\theta_t}(z) \mu_t(z) dz, \quad (20)$$

$$\frac{\partial \mu_t}{\partial t} = -\nabla_{\mu}\mathcal{F}(\theta_t, \mu_t), \quad \nabla_{\mu}\mathcal{F}(\theta_t, \mu_t) := -\nabla \cdot \left(\mu_t \left[P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t) \right] \right), \quad (21)$$

where $\nabla_{W_2}\mathcal{F}(\theta_t, \mu_t) = \nabla_z \log \left(\frac{\mu_t}{\pi_{\theta_t}} \right)$ denotes the Wasserstein gradient of $\mathcal{F}(\theta_t, \cdot)$ at μ_t . This gradient flow was obtained by replacing the Wasserstein gradient appearing in (4), (7) with its kernelized version, $P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$.

We first provide a result which quantifies the dissipation of the free energy along the trajectory of the continuous-time SVGD EM dynamics.

Proposition 1. *The dissipation of the free energy along the SVGD EM gradient flow (20) - (21) is given by*

$$\frac{d\mathcal{F}(\theta_t, \mu_t)}{dt} = -\int \|\nabla_{\theta} \log \pi_{\theta_t}(z)\|_{\mathbb{R}^{d_\theta}}^2 \mu_t(z) dz - \left\| S_{\mu_t} \nabla_z \log \left(\frac{\mu_t}{p_{\theta_t}(\cdot|x)} \right) \right\|_{\mathcal{H}_k^d}^2. \quad (22)$$

Proof. See App. B.1. □

Remark 2. *We can identify the second term in (22) as the Stein Fisher information of μ_t relative to the posterior $p_{\theta_t}(\cdot|x)$ Duncan et al. (2023); Korba et al. (2020), written*

$$I_{\text{Stein}}(\mu_t | p_{\theta_t}(\cdot|x)) := \left\| S_{\mu_t} \nabla_z \log \left(\frac{\mu_t}{p_{\theta_t}(\cdot|x)} \right) \right\|_{\mathcal{H}_k^d}^2. \quad (23)$$

This quantity is sometimes also referred to as the squared kernel Stein discrepancy (KSD).

Since both of the terms on the RHS in (22) are negative, Proposition 1 shows that the free energy decreases along the SVGD EM gradient flow. Under some additional assumption, we can actually establish convergence of the two terms on the RHS to zero. First, we will require the following rather mild regularity condition on the marginal likelihood; see also (Kuntz et al., 2023, Assumption 1).

Assumption 6. *The super-level sets of the marginal likelihood $p_{\theta}(x)$ are compact. That is, the set $\{\theta \in \Theta : p_{\theta}(x) \geq c\}$ is bounded for all $c \geq 0$.*

Proposition 2. *Let $(\theta_t)_{t \geq 0}$ and $(\mu_t)_{t \geq 0}$ be solutions of the SVGD EM gradient flow (20) - (21). Suppose that Assumptions 1, 2 and 6 hold. In addition, suppose that there exists a positive constant $C > 0$ such that $\int \|z\| \mu_t(z) dz < C$ for all $t \geq 0$. Then*

$$\lim_{t \rightarrow \infty} \left[\|\nabla_{\theta}\mathcal{F}(\theta_t, \mu_t)\|^2 + \left\| S_{\mu_t} \nabla_z \log \left(\frac{\mu_t}{\pi_{\theta_t}} \right) \right\|_{\mathcal{H}_k^d}^2 \right] = 0. \quad (24)$$

Proof. See App. B.1. □

Remark 3. We note that, as an alternative to Assumption 2, we could instead impose separate assumptions on the existence and boundedness of the Hessian H_{V_z} of the function $V_z : \Theta \rightarrow \mathbb{R}$, which maps $\theta \mapsto -\log \pi_\theta(z)$ for fixed $z \in \mathcal{Z}$; the Hessian H_{V_θ} of the function $V_\theta : \mathcal{Z} \rightarrow \mathbb{R}$ which maps $z \mapsto -\log \pi_\theta(z)$ for fixed $\theta \in \Theta$; and the matrix $H_{V_{\theta,z}} : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_\theta}$, which contains the mixed partial derivatives of V . Indeed, for certain results, e.g., Theorem 1, we only require boundedness of the ‘diagonal blocks’ of the Hessian H_V of the function $V : \Theta \times \mathcal{Z} \rightarrow \mathbb{R}$, $V : (\theta, z) \mapsto -\log \pi_\theta(z)$, which is strictly weaker than Assumption 2. For ease of presentation, in this case we choose to work directly with the Hessian H_V of V , as a function of both variables.

In order to establish exponential convergence along the SVGD EM gradient flow, we will require an additional condition, which characterizes the properties of \mathcal{F} around its equilibria. We assume a particular ‘gradient dominance’ condition, which represents a natural extension of the corresponding conditions used for Euclidean gradients flows - the Polyak-Łojasiewicz inequality - and for the SVGD gradient flow - the Stein log-Sobolev inequality Duncan et al. (2023); Korba et al. (2020).

Assumption 7. There exists $\lambda > 0$ such that \mathcal{F} satisfies the following gradient dominance condition

$$\mathcal{F}(\theta, \mu) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu) \leq \frac{1}{2\lambda} \|\nabla_{\mathcal{U}} \mathcal{F}\|_{\mathcal{U}}^2. \quad (25)$$

Proposition 3. Assume that Assumption 7 holds. Then the free energy $\mathcal{F}(\theta, \mu)$ decreases exponentially fast along the SVGD EM gradient flow. In particular,

$$\mathcal{F}(\theta_t, \mu_t) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu) \leq e^{-2\lambda t} [\mathcal{F}(\theta_0, \mu_0) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu)]. \quad (26)$$

Proof. See App. B.1. □

A.2 SVGD EM: Discrete Time Results

We now consider the forward Euler discretisation of the dynamics in (20) - (21), as given in (11) - (12). For convenience, we recall these update equations again now:

$$\theta_{t+1} = \theta_t + \gamma \int \nabla_\theta \log \pi_{\theta_t}(z) d\mu_t(z) \quad (27)$$

$$\mu_{t+1} = \left(\text{id} + \gamma \int [\nabla_z \log \pi_{\theta_{t+1}}(z) k(z, \cdot) + \nabla_z k(z, \cdot)] \mu_t(z) dz \right)_{\#} \mu_t. \quad (28)$$

The updates in (27) - (28) represent the population limit of SVGD EM (Alg. 1). In Theorem 1, we established a descent lemma, guaranteeing that, given a suitable choice of the learning rate γ , the free energy decreases at each iteration of the SVGD EM algorithm. As a corollary to this result, we now obtain a discrete time convergence rate for the average of $\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2$ and $\|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2$. In particular, we have the following result.

Corollary 1. Let $\alpha > 1$, and $\gamma \leq \min\left(\gamma^*, \frac{2}{M}, \frac{2}{(M+\alpha^2)B^2}\right)$. Then, defining $c_\gamma = \gamma \left(1 - \frac{[M+(M+\alpha^2)B^2]\gamma}{2}\right)$, the discrete-time, population-limit, SVGD EM updates in (11) - (12) satisfy

$$\min_{t=1, \dots, T} \left(\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \right) \quad (29)$$

$$\leq \frac{1}{T} \sum_{t=1}^T \left(\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \right) \leq \frac{\mathcal{F}(\theta_0, \mu_0) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu)}{c_\gamma T}. \quad (30)$$

Proof. See App. B.3. □

A.3 Coin EM

In this section, we outline how to establish convergence of the population-limit of an ‘ideal’ version of the Coin EM algorithm (Alg. 3). In particular, for this result, we will consider a version of the Coin EM algorithm in which the outcomes observed by the two gamblers are $c_t^{(1)} = -\nabla_\theta \mathcal{F}(\theta_t, \mu_t)$ and $c_t^{(2)} = -\nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$, rather than $c_t^{(1)} = -\nabla_\theta \mathcal{F}(\theta_t, \mu_t)$ and $c_t^{(2)} = -\mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$ as in the original Coin EM algorithm (Alg. 2).

Algorithm 3 Coin EM (Ideal Algorithm, Population Limit)

- 1: **Input:** number of iterations T , number of particles N , initial particle $z_0 \sim \mu_0$, initial θ_0 , target π , kernel k .
- 2: **for** $t = 1, \dots, T$ **do**
- 3:

$$\theta_t = \theta_0 - \frac{\sum_{s=1}^{t-1} \nabla_{\theta} \mathcal{F}(\theta_s, \mu_s)}{t} \left(1 - \sum_{s=1}^{t-1} \langle \nabla_{\theta} \mathcal{F}(\theta_s, \mu_s), \theta_s - \theta_0 \rangle\right), \quad (31)$$

$$z_t = z_0 - \frac{\sum_{s=1}^{t-1} \mathcal{P}_{\mu_s} \nabla_{W_2} \mathcal{F}(\theta_s, \mu_s)(z_s)}{t} \left(1 - \sum_{s=1}^{t-1} \langle \mathcal{P}_{\mu_s} \nabla_{W_2} \mathcal{F}(\theta_s, \mu_s)(z_s), z_s - z_0 \rangle\right), \quad (32)$$

- 4: **return** θ_T and z_T .
-

In order to establish the convergence of this algorithm, we will require an additional technical assumption on the transport maps defined by (32) in Alg. 3. We emphasize that it remains a challenging open question how to verify this assumption in practice; see also the discussion in Sharrock and Nemeth (2023). We leave this question to future work.

Assumption 8. *There exists a transport map $T_{\mu_0}^{p_{\theta^*}(\cdot|x)} : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$ from μ_0 to $p_{\theta^*}(\cdot|x)$, where $\theta^* = \arg \max_{\theta \in \Theta} p_{\theta}(x)$, and a constant $K > 0$, such that*

$$\sum_{t=1}^T \left\| t_{\mu_t}^{p_{\theta^*}(\cdot|x)} \circ z_t - T_{\mu_0}^{p_{\theta^*}(\cdot|x)} \right\|_{L^2(\mu_0)} \leq K\sqrt{T} \ln [\text{poly}(T)], \quad (33)$$

where, for each $t \in [T]$, $t_{\mu_t}^{p_{\theta^*}(\cdot|x)} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the optimal transport map from μ_t to $p_{\theta^*}(\cdot|x)$ (e.g., Brenier, 1991), and $z_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the transport map from μ_0 to μ_t as defined by (32) in Alg. 3.

This assumption represents an extension of Assumption B.1' in Sharrock and Nemeth (2023). It can, in some sense, be interpreted as a bound on the sum of the distances between some fixed transport map from μ_0 to $p_{\theta^*}(\cdot|x)$ - e.g., the optimal transport map $t_{\mu_0}^{p_{\theta^*}(\cdot|x)}$ - and the maps $(t_{\mu_t}^{\pi} \circ z_t)_{t \in [T]}$ from μ_0 to $p_{\theta^*}(\cdot|x)$, which first transport μ_0 to $(\mu_t)_{t \in [T]}$ according to the transport maps $(z_t)_{t \in [T]}$ defined by Alg. 3, and then map $(\mu_t)_{t \in [T]}$ to π via the optimal transport maps $(t_{\mu_t}^{p_{\theta^*}(\cdot|x)})_{t \in [T]}$.

We are now ready to provide a full statement of Theorem 2.

Theorem 2. *Assume that Assumptions 4, 5, and 8 hold. Then the marginal likelihood $\theta \mapsto p_{\theta}(x)$ admits a unique maximizer $\theta^* = \arg \max_{\theta \in \Theta} p_{\theta}(x)$. In addition, writing $\mu^* = p_{\theta^*}(\cdot|x)$, it holds for all $T \geq 1$ that*

$$\begin{aligned} \mathcal{F} \left(\frac{1}{T} \sum_{t=1}^T \theta_t, \frac{1}{T} \sum_{t=1}^T \mu_t \right) - \mathcal{F}(\theta^*, \mu^*) &\leq \frac{L}{T} \left[2 + K\sqrt{T} \ln [\text{poly}(T)] + \|\theta^* - \theta_0\| \sqrt{T \ln (1 + 24T^2 \|\theta^* - \theta_0\|^2)} \right] \\ &\quad + \int_{\mathbb{R}^d} \left\| T_{\mu_0}^{p_{\theta^*}(\cdot|x)}(z) - z \right\| \sqrt{T \ln (1 + 24T^2 \|T_{\mu_0}^{p_{\theta^*}(\cdot|x)}(z) - z\|^2)} \mu_0(z) dz \end{aligned} \quad (34)$$

B PROOFS OF THEORETICAL RESULTS

B.1 Proof of Propositions 1, 2, and 3.

Proof of Proposition 1. Using differential calculus in the product space $\Theta \times \mathcal{P}_2(\mathcal{Z})$, and the chain rule, we have that

$$\begin{aligned} \frac{d\mathcal{F}(\theta_t, \mu_t)}{dt} &= \langle -\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t), \nabla_{\theta} \mathcal{F}(\theta_t, \mu_t) \rangle_{\mathbb{R}^{d_{\theta}}} + \langle P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t), \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t) \rangle_{L^2(\mu_t)} \\ &= -\|\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_{\theta}}}^2 - \|S_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \end{aligned} \quad (35)$$

where in the second line we have used the fact that, given $\mu \in \mathcal{P}_2(\mathcal{Z})$, and functions $f, g \in L^2(\mu)$, $\mathcal{H}_k^{d_z}$, it holds that $\langle f, \iota g \rangle_{L^2(\mu)} = \langle \iota^* f, g \rangle_{\mathcal{H}_k^{d_z}} = \langle S_{\mu} f, g \rangle_{\mathcal{H}_k^{d_z}}$, since the adjoint of the inclusion $\iota : \mathcal{H} \rightarrow L^2(\mu)$ is $\iota^* = S_{\mu}$. To

obtain the first term on the RHS of (22), we can now just substitute the expression for $\nabla_{\theta}\mathcal{F}(\theta_t, \mu_t)$ from (20) into the first term on the RHS of (35). For the remaining term, recalling the definition of $\pi_{\theta}(z)$, namely, $\pi_{\theta}(z) := p_{\theta}(z, x) = p_{\theta}(z|x)p_{\theta}(x)$, we have

$$\nabla_z \log \left(\frac{\mu_t}{\pi_{\theta_t}} \right) = \nabla_z \log \left(\frac{\mu_t}{p_{\theta_t}(\cdot|x)p_{\theta_t}(x)} \right) = \nabla_z \log \left(\frac{\mu_t}{p_{\theta_t}(\cdot|x)} \right) \quad (36)$$

where the final equality follows from the fact that $p_{\theta_t}(x)$ is independent of z . Substituting this into (35) completes the proof. \square

Proof of Proposition 2. We use similar arguments to those in the proofs of (Kuntz et al., 2023, Theorem 3) and (Korba et al., 2020, Proposition 8), adapted appropriately to our setting. For notational convenience, let us define

$$I(\theta_t, \mu_t) := \|\nabla_{\theta}\mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_{\theta}}}^2 + \|S_{\mu_t}\nabla_{W_2}\mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2. \quad (37)$$

We start by establishing that, under the stated assumptions, there exists a positive constant $\beta > 0$ which guarantees that

$$\left| \frac{dI(\theta_t, \mu_t)}{dt} \right| \leq \beta I(\theta_t, \mu_t). \quad (38)$$

We will then show that this implies convergence of $I(\theta_t, \mu_t)$ to zero. To prove (38), we first compute

$$\frac{dI(\theta_t, \mu_t)}{dt} = \underbrace{\frac{d}{dt} \|\nabla_{\theta}\mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_{\theta}}}^2}_{I_t^{(1)}(\theta_t, \mu_t)} + \underbrace{\frac{d}{dt} \|S_{\mu_t}\nabla_{W_2}\mathcal{F}(\theta_t, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2}_{I_t^{(2)}(\theta_t, \mu_t)} \quad (39)$$

We begin with $I_t^{(1)}(\theta_t, \mu_t)$. Let us define $v_t = \nabla_{\theta}\mathcal{F}(\theta_t, \mu_t) = -\int \nabla_{\theta} \log \pi_{\theta_t}(z) \mu_t(z) dz$ and $w_t = S_{\mu_t} \nabla_{W_2}\mathcal{F}(\theta_t, \mu_t) = S_{\mu_t} \nabla \log(\frac{\mu_t}{\pi_{\theta_t}})$. We then have

$$I_t^{(1)}(\theta_t, \mu_t) = \frac{d}{dt} \|v_t\|_{\mathbb{R}^{d_{\theta}}}^2 = 2 \langle v_t, \frac{d}{dt} v_t \rangle_{\mathbb{R}^{d_{\theta}}}. \quad (40)$$

We now need to obtain $\frac{d}{dt} v_t$. Using the chain rule, and then integration by parts, we have

$$\frac{d}{dt} v_t = - \int \frac{d}{dt} [\nabla_{\theta} \log \pi_{\theta_t}(z) \mu_t(z)] dz \quad (41)$$

$$= - \int \frac{d}{dt} [\nabla_{\theta} \log \pi_{\theta_t}(z)] \mu_t(z) dz - \int \nabla_{\theta} \log \pi_{\theta_t}(z) \frac{\partial \mu_t}{\partial t}(z) dz \quad (42)$$

$$= - \int \nabla_{\theta}^2 \log \pi_{\theta_t}(z) \mu_t(z) \frac{\partial \theta_t}{\partial t} dz - \int \nabla_{\theta} \log \pi_{\theta_t}(z) \nabla_z \cdot (w_t(z) \mu_t(z)) dz \quad (43)$$

$$= \int \nabla_{\theta}^2 \log \pi_{\theta_t}(z) v_t \mu_t(z) dz + \int \nabla_{\theta} \nabla_z \log \pi_{\theta_t}(z) w_t(z) \mu_t(z) dz. \quad (44)$$

It follows, in particular, that

$$I_t^{(1)}(\theta_t, \mu_t) = -2 \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_{\theta}} \int v_t^i [\nabla_{\theta}^2 \log \pi_{\theta_t}(z)]_{ij} v_t^j \mu_t(z) dz + 2 \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_z} \int v_t^i [\nabla_{\theta} \nabla_z \log \pi_{\theta_t}(z)]_{ij} w_t^j(z) \mu_t(z) dz. \quad (45)$$

For the second term, we will work component-wise. First, using the notation $w_t = (w_t^1, \dots, w_t^{d_z})$, we have $\|w_t\|_{\mathcal{H}_k}^2 = \sum_{j=1}^{d_z} \|w_t^j\|_{\mathcal{H}_k}^2$. Thus, in particular, $I_t^{(2)}(\theta_t, \mu_t) = \frac{d}{dt} \sum_{j=1}^{d_z} \|w_t^j\|_{\mathcal{H}_k}^2 = 2 \sum_{j=1}^{d_z} \langle w_t^j, \frac{d}{dt} w_t^j \rangle_{\mathcal{H}_k}$. It remains to compute the components $\frac{d}{dt} w_t^j$, for $j \in [d_z]$. Starting from the definition, using integration by parts, the chain

rule, and then integration by parts again, we have

$$\frac{d}{dt} w_t^j(z) = \frac{d}{dt} \int k(z', z) \partial_{z'_i} \log \left(\frac{\mu_t}{\pi_{\theta_t}} \right) (z') \mu_t(z') dz' \quad (46)$$

$$= - \int \frac{d}{dt} \left[\left[\partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z) + \partial_{z'_j} k(z', z) \right] \mu_t(z') \right] dz' \quad (47)$$

$$= - \int \left\langle \nabla_{\theta} \partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z), \frac{\partial \theta_t}{\partial t} \right\rangle_{\mathbb{R}^{d_{\theta}}} \mu_t(z') dz' \\ - \int \left[\partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z) + \partial_{z'_j} k(z', z) \right] \frac{\partial}{\partial t} \mu_t(z') dz' \quad (48)$$

$$= \int \left\langle \nabla_{\theta} \partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z), v_t \right\rangle_{\mathbb{R}^{d_{\theta}}} \mu_t(z') dz' \\ + \int \left\langle \nabla_{z'} \left[\partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z) + \partial_{z'_j} k(z', z) \right], w_t(z') \right\rangle_{\mathbb{R}^{d_z}} \mu_t(z') dz' \quad (49)$$

$$= \int \left\langle \nabla_{\theta} \partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z), v_t \right\rangle_{\mathbb{R}^{d_{\theta}}} \mu_t(z') dz' \\ + \int \sum_{i=1}^{d_z} \left[\partial_{z'_i} \partial_{z'_j} \log \pi_{\theta_t}(z') k(z', z) + \partial_{z'_j} \log \pi_{\theta_t}(z') \partial_{z'_i} k(z', z) \right. \\ \left. + \partial_{z'_i} \partial_{z'_j} k(z', z) \right] w_t^j(z') \mu_t(z') dz'. \quad (50)$$

Now, using the fact that each component of w_t^j is an elements of the RKHS \mathcal{H}_k , and thus satisfies the reproducing property $w_t^j(z) = \langle w_t^j, k(z, \cdot) \rangle_{\mathcal{H}_k}$, we can rewrite the previous display as

$$I_t^{(2)}(\theta_t, \mu_t) = 2 \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_z} \int v_t^i [\nabla_{\theta} \nabla_z \log \pi_{\theta_t}(z)]_{ij} w_t^j(z) \mu_t(z) dz \\ + 2 \sum_{i=1}^d \sum_{j=1}^{d_z} \int \left[\partial_{z_i} \partial_{z_j} \log \pi_{\theta_t}(z) w_t^j(z) + \partial_{z_i} \log \pi_{\theta_t}(z) \partial_{z_j} w_t^j(z) + \partial_{z_j} \partial_{z_i} w_t^j(z) \right] w_t^j(z) \mu_t(z) dz. \quad (51)$$

Substituting (45) and (51) into (39), before once more making use of the reproducing property, we thus have

$$\frac{dI(\theta_t, \mu_t)}{dt} = 2 \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_{\theta}} \int v_t^i [-\nabla_{\theta}^2 \log \pi_{\theta_t}(z)]_{ij} v_t^j \mu_t(z) dz \\ + 2 \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_z} \int v_t^i [\nabla_{\theta} \nabla_z \log \pi_{\theta_t}(z)]_{ij} w_t^j(z) \mu_t(z) dz \\ + 2 \sum_{i=1}^{d_z} \sum_{j=1}^{d_z} \int \left[\partial_{z_i} \partial_{z_j} \log \pi_{\theta_t}(z) w_t^j(z) + \partial_{z_i} \log \pi_{\theta_t}(z) \partial_{z_j} w_t^j(z) + \partial_{z_j} \partial_{z_i} w_t^j(z) \right] w_t^j(z) \mu_t(z) dz \quad (52)$$

$$= \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_{\theta}} v_t^i A_t^{ij} v_t^j + \sum_{i=1}^{d_{\theta}} \sum_{j=1}^{d_z} v_t^i \langle B_t^{ij}, w_t^j \rangle_{\mathcal{H}_k} + \sum_{i=1}^{d_z} \sum_{j=1}^{d_z} \langle w_t^i, C_t^{ij} w_t^j \rangle_{\mathcal{H}_k}, \quad (53)$$

where in the final line we have defined

$$A_t^{ij} = 2 \int [-\partial_{\theta_i} \partial_{\theta_j} \log \pi_{\theta_t}(z)] \mu_t(dz) \quad (54)$$

$$B_t^{ij} = 2 \int k(z, \cdot) \partial_{\theta_i} \partial_{z_i} \log \pi_{\theta_t}(z) \mu_t(z) dz \quad (55)$$

$$\begin{aligned} C_t^{ij} &= 2 \int k(z, \cdot) \otimes k(z', \cdot) \partial_{z_i} \partial_{z_j} \log \pi(z) \mu_t(z) \mu_t(z') dz dz' \\ &\quad + 2 \int \partial_{z_i} k(z, \cdot) \otimes k(z', \cdot) \partial_{z_i} \log \pi_{\theta_t}(z) \mu_t(z) dz \mu_t(z') dz' \\ &\quad + 2 \int \partial_{z_i} k(z, \cdot) \otimes \partial_{z_j} k(z', \cdot) \mu_t(z) \mu_t(z') dz dz'. \end{aligned} \quad (56)$$

We now need to bound each of these terms in an appropriate sense. In particular, if we can show that $\|A_t^{ij}\|_{\text{op}} \leq D_1$, $\|B_t^{ij}\|_{\mathcal{H}_k} \leq D_2$, and $\|C_t^{ij}\|_{\text{HS}} \leq D_3$, for all $t \geq 0$, where $\|\cdot\|_{\text{op}}$ denotes the operator norm, and $\|\cdot\|_{\text{HS}}$ denotes the Hilbert-Schmidt norm, then it follows immediately that

$$\begin{aligned} \left| \frac{dI(\theta_t, \mu_t)}{dt} \right| &\leq D_1 \sum_{i=1}^{d_\theta} \|v_t^i\|^2 + D_2 d_\theta^{\frac{1}{2}} d_z^{\frac{1}{2}} \left(\sum_{i=1}^{d_\theta} \|v_t^i\|^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^{d_z} \|w_t^i\|_{\mathcal{H}_k}^2 \right)^{\frac{1}{2}} + D_3 d_z \sum_{i=1}^{d_z} \|w_t^i\|_{\mathcal{H}_k}^2 \\ &\leq D \left(\|v_t\|_{\mathbb{R}^{d_\theta}}^2 + 2\|v_t\|_{\mathbb{R}^{d_\theta}} \|w_t\|_{\mathcal{H}_k^{d_z}} + \|w_t\|_{\mathcal{H}_k^{d_z}}^2 \right) \leq 2D \left(\|v_t\|_{\mathbb{R}^{d_\theta}}^2 + \|w_t\|_{\mathcal{H}_k^{d_z}}^2 \right) := \beta I(\theta_t, \mu_t), \end{aligned} \quad (57)$$

where in the first line we have used the Cauchy-Schwarz inequality, in the second line we have defined the constant $D = \max(D_1, \frac{1}{2} D_2 d_\theta^{\frac{1}{2}} d_z^{\frac{1}{2}}, D_3 d_z)$, and in the final line we have used $a^2 + 2ab + b^2 = (a+b)^2 \leq 2(a^2 + b^2)$, and set $\beta = 2D$. It remains to obtain the bounds on A_t^{ij} , B_t^{ij} , and C_t^{ij} .

For the first term, we just require the boundedness of the Hessian $H_{V_z} : \mathbb{R}^{d_\theta} \rightarrow \mathbb{R}^{d_\theta}$ of the function $V_z : \theta \mapsto -\log \pi_\theta(z)$, for each $z \in \mathcal{Z}$, which is guaranteed by Assumption 2. In particular, by Assumption 2, we certainly have that $\|H_{V_z}\|_{\text{op}} \leq M$, from which it immediately follows that $\|A_t^{ij}\|_{\text{op}} \leq 2M$. For the second term, once again using Assumption 2, as well as the bound of the kernel (Assumption 1), we have

$$\|B_t^{ij}\|_{\mathcal{H}_k} \leq 2 \int \|k(z, \cdot)\|_{\mathcal{H}_k} |\partial_{\theta_i} \partial_{z_i} \log \pi_{\theta_t}(z)| \mu_t(z) dz \leq BM. \quad (58)$$

For the final term, we argue similarly to in (Korba et al., 2020, Proposition 8). In particular, using the boundedness of the Hessian $H_{V_\theta} : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$ of the function $V_\theta : z \mapsto -\log \pi_\theta(z)$ for each $\theta \in \Theta$, which is an immediate consequence of Assumption 2, and the boundedness of the kernel and its gradient (Assumption 1), we have

$$\begin{aligned} \|C_t^{ij}\|_{\text{HS}} &\leq 2 \int \|k(z, \cdot)\|_{\mathcal{H}_k} |\partial_{z_i} \partial_{z_j} \log \pi_{\theta_t}(z)| \mu_t(z) dz \|k(z', \cdot)\|_{\mathcal{H}_k} \mu_t(z') dz' \\ &\quad + 2 \int \|\partial_{z_i} k(z, \cdot)\|_{\mathcal{H}_k} |\partial_{z_i} \log \pi_{\theta_t}(z)| \mu_t(z) dz \int \|k(z', \cdot)\|_{\mathcal{H}_k} \mu_t(z') dz' \\ &\quad + 2 \left(\int \|k(z, \cdot)\|_{\mathcal{H}_k} \mu_t(z) dz \right)^2 \leq 2B^2 \left(M + \int |\partial_{z_i} \log \pi_{\theta_t}(z)| \mu_t(z) dz + 1 \right). \end{aligned} \quad (59)$$

To bound the remaining integral, we use the fact that $|\log \pi_\theta(z)| \leq |\partial_{z_i} \log \pi_\theta(0)| + M\|z\|$, which follows from Assumption 2. It follows that

$$\int |\partial_{z_i} \log \pi_{\theta_t}(z)| \mu_t(dz) \leq \int [|\partial_{z_i} \log \pi_{\theta_t}(0)| + M\|z\|] \mu_t(z) dz \leq |\partial_{z_i} \log \pi_{\theta_t}(0)| + MC. \quad (60)$$

Substituting (60) into (59), we thus have that $\|C_t^{ij}\|_{\text{HS}} \leq 2B^2(1 + M + CM + |\partial_{z_i} \log \pi_{\theta_t}(0)|)$. Thus, setting $D_1 = 2M$, $D_2 = BM$, $D_3 = 2B^2(1 + M + CM + |\partial_{z_i} \log \pi_{\theta_t}(0)|)$, and using the argument in (57), we have established the result in (38).

It remains to establish that $I(\theta_t, \mu_t) \rightarrow 0$. Once more, we will utilize some of the ideas from the proofs of (Kuntz et al., 2023, Theorem 3) and (Korba et al., 2020, Proposition 8). We begin by observing that the free energy can

be written as

$$\mathcal{F}(\theta, \mu) = \int \nabla_z \log \left(\frac{\mu}{\pi_\theta} \right) \mu(z) dz = \int \log \left(\frac{\mu}{p_\theta(\cdot|x)p_\theta(x)} \right) \mu(z) dz = \underbrace{\int \log \left(\frac{\mu_t}{p_{\theta_t}(\cdot|x)} \right) \mu(z) dz}_{\text{KL}(\mu_t|p_{\theta_t}(\cdot|x))} - \log p_\theta(x). \quad (61)$$

Rearranging (61), using the non-negativity of the KL divergence, and finally the fact that the free energy dissipates along the flow of the SVGD EM dynamics (Proposition 1), it then follows that

$$\log p_{\theta_t}(x) = -\mathcal{F}(\theta_t, \mu_t) + \text{KL}(\mu_t|p_{\theta_t}(\cdot|x)) \geq \mathcal{F}(\theta_t, \mu_t) \geq \mathcal{F}(\theta_0, \mu_0). \quad (62)$$

Under Assumption 6, the set $\Theta_{\mathcal{F}_0} = \{\theta \in \Theta : \log p_\theta(x) \geq \mathcal{F}(\theta_0, \mu_0)\}$ is compact. This, along with (62), immediately implies that $(\theta_t)_{t \geq 0}$ is relatively compact. We now return to (61). In particular, once more rearranging this equation, we have that, for each $t \geq 0$,

$$\inf_{\theta \in \Theta_{\mathcal{F}_0}} \text{KL}(\mu_t|p_\theta(\cdot|x)) \leq \text{KL}(\mu_t|p_{\theta_t}(\cdot|x)) = \mathcal{F}(\theta_t, \mu_t) + \log p_{\theta_t}(x) \quad (63)$$

$$\leq \mathcal{F}(\theta_0, \mu_0) + \log p_{\theta_t}(x) \leq \mathcal{F}(\theta_0, \mu_0) + \sup_{\theta \in \Theta_{\mathcal{F}_0}} \log p_\theta(x). \quad (64)$$

It thus follows, using also the fact that the KL divergence has compact sub-level sets in the weak topology (van Erven and Harremos, 2014, Theorem 20), that the family $(\mu_t)_{t \geq 0}$ is weakly relatively compact. Based on these two observations, and also the (weak) continuity of $I(\theta, \mu)$, we can conclude that $\sup_{t \geq 0} I(\theta_t, \mu_t) < \infty$. Thus, using the bound (38) that we established earlier in the proof, we have that $|\frac{d}{dt} I(\theta_t, \mu_t)| \leq K$ for some $K > 0$.

Finally, we are ready to show that $I(\theta_t, \mu_t) \rightarrow 0$. We follow closely the final part of the proof of (Korba et al., 2020, Proposition 8), which we can now apply almost verbatim. For the interested reader, we also provide the details here. We will argue by contradiction. In particular, suppose that $I(\theta_t, \mu_t)$, so that there exists a sequence $t_m \rightarrow \infty$ such that $I(\theta_{t_m}, \mu_{t_m}) > \varepsilon > 0$. In addition, since $|\frac{d}{dt} I(\theta_t, \mu_t)|$ is bounded, it is uniformly K Lipschitz in time. Thus, there exists a sequence of intervals J_m of length $\frac{m}{K}$, and centered at t_m , such that $I(\theta_t, \mu_t) \geq \frac{\varepsilon}{2}$ for all $t \in J_m$. Finally, integrating the dissipation over $s \in [0, t]$, we have

$$\mathcal{F}(\theta_0, \mu_0) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu) \geq \mathcal{F}(\theta_0, \mu_0) - \mathcal{F}(\mu_t, \theta_t) = \int_0^t I(\theta_s, \mu_s) ds \geq \sum_{m: t_m \leq t} \frac{\varepsilon^2}{2K}, \quad (65)$$

which diverges as $t \rightarrow \infty$ since the subsequence $t_m \rightarrow \infty$. But this is contradiction, since $\mathcal{F}(\theta_0, \mu_0) < \infty$. This completes the proof. \square

Proof of Proposition 3. Suppose we write $\mathcal{F}^* = \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu)$. Proceeding almost identically to the start of the proof of Proposition 1, we have

$$\frac{d}{dt} (\mathcal{F}(\theta_t, \mu_t) - \mathcal{F}^*) \quad (66)$$

$$= \langle -\nabla_\theta \mathcal{F}(\theta_t, \mu_t), \nabla_\theta \mathcal{F}(\theta_t, \mu_t) \rangle_{\mathbb{R}^{d_\theta}} + \langle -P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t), \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t) \rangle_{L^2(\mu_t)} \quad (67)$$

$$= -\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 - \|S_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \quad (68)$$

$$\leq -2\lambda (\mathcal{F}(\theta_t, \mu_t) - \mathcal{F}^*). \quad (69)$$

where in the final line we have used the gradient dominance condition (Assumption 7). The conclusion now follows straightforwardly via Grönwall's inequality. \square

B.2 Proof of Theorem 1.

Proof. For fixed $t \in \mathbb{N}_0$, consider the following decomposition

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_t, \mu_t) = \underbrace{\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_{t+1}, \mu_t)}_I + \underbrace{\mathcal{F}(\theta_{t+1}, \mu_t) - \mathcal{F}(\theta_t, \mu_t)}_{II}. \quad (70)$$

We will deal with (I) and (II) in turn, beginning with (II). First, for each $\tau \geq 0$, and for fixed $t \in \mathbb{N}_0$, let $\theta_\tau = \theta_t - \tau \nabla_\theta \mathcal{F}(\theta_t, \mu_t)$. We then have $\theta_0 = \theta_t$ and $\theta_\gamma = \theta_t - \gamma \nabla_\theta \mathcal{F}(\theta_t, \mu_t) = \theta_{t+1}$. We also have that $\dot{\theta}_\tau = -\nabla_\theta \mathcal{F}(\theta_t, \mu_{t+1})$ and $\ddot{\theta}_\tau = 0$, where here we emphasize that \cdot and $\ddot{\cdot}$ denote derivatives with respect to the continuous $\tau \in \mathbb{R}_+$, rather than the discrete $t \in \mathbb{N}_0$.

In addition, suppose we let $g(\tau) = \mathcal{F}(\theta_\tau, \mu_t)$. We then have $g(0) = \mathcal{F}(\theta_t, \mu_t)$ and $g(\gamma) = \mathcal{F}(\theta_{t+1}, \mu_t)$. In addition, using a Taylor expansion, we have that

$$g(\gamma) = g(0) + \gamma g'(0) + \int_0^\gamma (\gamma - \tau) g''(\tau) d\tau. \quad (71)$$

Let us identify the remaining terms in this expansion. Using the chain rule, and in the second line also the fact that $\ddot{\theta}_\tau = 0$, we have that

$$g'(\tau) = \langle \dot{\theta}_\tau, \nabla_\theta \mathcal{F}(\theta_\tau, \mu_t) \rangle_{\mathbb{R}^{d_\theta}} = -\langle \nabla_\theta \mathcal{F}(\theta_t, \mu_t), \nabla_\theta \mathcal{F}(\theta_\tau, \mu_t) \rangle_{\mathbb{R}^{d_\theta}} \quad (72)$$

$$g''(\tau) = \langle \dot{\theta}_\tau, \nabla_\theta^2 \mathcal{F}(\theta_\tau, \mu_t) \dot{\theta}_\tau \rangle_{\mathbb{R}^{d_\theta}} = \langle \nabla_\theta \mathcal{F}(\theta_t, \mu_t), \text{Hess}_\theta(\mathcal{F})(\theta_\tau, \mu_t) \nabla_\theta \mathcal{F}(\theta_t, \mu_t) \rangle_{\mathbb{R}^{d_\theta}} \quad (73)$$

where $\text{Hess}_\theta(\mathcal{F})$ denotes the Hessian matrix of $\mathcal{F}(\cdot, \mu_t)$. Thus, $g'(0) = -\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2$, and $g''(\tau) \leq M \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2$, the latter by Assumption 2. Putting everything together, we thus have that

$$\mathcal{F}(\theta_{t+1}, \mu_t) \leq \mathcal{F}(\theta_t, \mu_t) - \gamma \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + M \int_0^\gamma (\gamma - \tau) \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 d\tau \quad (74)$$

$$\leq \mathcal{F}(\theta_t, \mu_t) - \gamma \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \frac{M\gamma^2}{2} \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 \quad (75)$$

which implies, in particular, that

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_t, \mu_t) \leq -\gamma \left(1 - \frac{M\gamma}{2}\right) \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2. \quad (76)$$

We will use a very similar argument to obtain an upper bound on (I). Here, we follow rather closely the proofs of (Korba et al., 2020, Proposition 5) and (Salim et al., 2022, Proposition 3.1), adapted appropriately to our setting. In the interest of completeness, we provide this argument in full.

For $\tau \geq 0$, let $\psi_\tau = \text{id} - \tau \xi$, where $\xi = \mathcal{P}_\mu g \in L^2(\mu)$ $\mu \in \mathcal{P}_2(\mathcal{Z})$, for some $g \in L^2(\mu)$, and where id denotes the identity operator. In addition, let $\rho_\tau = (\psi_\tau)_\# \mu_t$. By definition, we then have that $\rho_0 = \mu_t$ and $\rho_\gamma = \mu_{t+1}$. Finally, let $\xi' = S_\mu g$, with $\xi(z) = \xi'(z)$ for all $z \in \mathcal{Z}$. We begin by observing that, for each $(\theta, z) \in \Theta \times \mathcal{Z}$, the reproducing property and the Cauchy-Schwarz inequality yield the bounds

$$\|\xi(z)\|^2 = \sum_{i=1}^{d_z} \langle k(z, \cdot), \xi'_i \rangle_{\mathcal{H}_k}^2 \leq \|k(z, \cdot)\|_{\mathcal{H}_k}^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2 \leq B^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2 \quad (77)$$

$$\|J\xi(z)\|_{\text{HS}}^2 = \sum_{i,j=1}^{d_z} \langle \partial_{z_j} k(z, \cdot), \xi'_i \rangle_{\mathcal{H}_k}^2 \leq \sum_{i,j=1}^{d_z} \|\partial_{z_j} k(z, \cdot)\|_{\mathcal{H}_k}^2 \|\xi'_i\|_{\mathcal{H}_k}^2 = \|\nabla k(z, \cdot)\|_{\mathcal{H}_k}^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2 \leq B^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2, \quad (78)$$

where in each case the final display follows from Assumption 1. Suppose, in addition, that for some $\alpha > 1$, the step size γ satisfies

$$\gamma \|\xi'\|_{\mathcal{H}_k^{d_z}} \leq \frac{\alpha - 1}{\alpha B}. \quad (79)$$

It then follows from the previous bound that, for any $\tau \in [0, \gamma]$,

$$\|\tau J\xi(z)\|_{\text{op}} \leq \|\tau J\xi(z)\|_{\text{HS}} \leq \gamma B \|\xi'\|_{\mathcal{H}_k^{d_z}} \leq \frac{\alpha - 1}{\alpha} < 1. \quad (80)$$

where J denotes the Jacobian matrix. This establishes that $J(\psi_\tau)(z) = \text{id} - \tau J\xi(z)$ is regular for all z , and that ψ_τ is a diffeomorphism for each $\tau \in [0, \gamma]$. In addition, we also have the bound

$$\|(J\psi_\tau)^{-1}(z)\|_{\text{op}} \leq \sum_{i=0}^{\infty} \|\tau J\xi(z)\|_{\text{op}}^i \leq \sum_{i=0}^{\infty} \left(\frac{\alpha - 1}{\alpha}\right)^i = \alpha. \quad (81)$$

Now, let us define $h(\tau) = \mathcal{F}(\theta_{t+1}, \rho_\tau)$. Using the definition above, we have that $h(0) = \mathcal{F}(\theta_{t+1}, \mu_t)$ and $h(\gamma) = \mathcal{F}(\theta_{t+1}, \mu_{t+1})$. Now, via a Taylor expansion,

$$h(\gamma) = h(0) + \gamma h'(0) + \int_0^\gamma (\gamma - \tau) h''(\tau) d\tau. \quad (82)$$

Similar to before, we need to identify the final two terms. We first recall, from (Villani, 2003, Theorem 5.34), that the velocity field which rules the time evolution of ρ_τ is given by $w_\tau \in L^2(\rho_\tau)$, defined according to $w_\tau(z) = -\xi(\psi_\tau^{-1}(z))$. Then, by an appropriate version of the chain rule (e.g. Villani, 2003, Section 8.2), one can show that

$$h'(0) = -\langle \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t), \xi \rangle_{L^2(\mu_t)} = -\langle \mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t), \xi' \rangle_{\mathcal{H}_k^{d_z}} \quad (83)$$

$$h''(\tau) = \underbrace{\mathbb{E}_{z \sim \rho_\tau} [\langle w_\tau(z), \text{Hess}_z(-\log \pi_{\theta_{t+1}}(z)) w_\tau(z) \rangle]}_{h_1(\tau)} + \underbrace{\mathbb{E}_{z \sim \rho_\tau} [\|J(w_\tau(z))\|_{\text{HS}}^2]}_{h_2(\tau)}, \quad (84)$$

It remains to bound these two terms. Recalling that $\rho_\tau = (\psi_\tau)_\# \mu_t$ and $w_\tau(z) = -\xi(\psi_\tau^{-1}(z))$, and using Assumption 2, we have

$$h_1(\tau) = \mathbb{E}_{z \sim \rho_\tau} [\langle w_\tau(z), \text{Hess}(-\log \pi_{\theta_{t+1}}(z)) w_\tau(z) \rangle] \quad (85)$$

$$= \mathbb{E}_{z \sim \rho_\tau} [\langle \xi(\psi_\tau^{-1}(z)), \text{Hess}(-\log \pi_{\theta_{t+1}}(z)) \xi(\psi_\tau^{-1}(z)) \rangle] \quad (86)$$

$$= \mathbb{E}_{z \sim \mu_t} [\langle \xi(z), \text{Hess}(-\log \pi_{\theta_{t+1}}(z)) \xi(z) \rangle] \quad (87)$$

$$\leq M \mathbb{E}_{z \sim \mu_t} [\|\xi(z)\|^2] \leq MB^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2, \quad (88)$$

where the final inequality follows from the bound in (77). For the remaining term, by the chain rule, we have $Jw_\tau(z) = J\xi(\psi_\tau^{-1}(z))(J\psi_\tau)^{-1}(\psi_\tau^{-1}(z))$. It follows that

$$h_2(\tau) = \mathbb{E}_{z \sim \rho_\tau} [\|J(w_\tau(z))\|_{\text{HS}}^2] \quad (89)$$

$$= \mathbb{E}_{z \sim \rho_\tau} [\|J\xi(\psi_\tau^{-1}(z))(J\psi_\tau)^{-1}(\psi_\tau^{-1}(z))\|_{\text{HS}}^2] \quad (90)$$

$$= \mathbb{E}_{z \sim \mu_t} [\|J\xi(z)(J\psi_\tau)^{-1}(z)\|_{\text{HS}}^2] \quad (91)$$

$$\leq \mathbb{E}_{z \sim \mu_t} [\|J\xi(z)\|_{\text{HS}}^2 \|(J\psi_\tau)^{-1}(z)\|_{\text{op}}^2] \leq \alpha^2 B^2 \|\xi'\|_{\mathcal{H}_k^{d_z}}^2, \quad (92)$$

where the final display follows from the bounds in (78) and (81). Clearly, we would like to apply these results with $\xi = \xi_t = \mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)$, $\xi' = \xi'_t = \mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)$, which corresponds to the SVGD EM update. In this case, substituting (83) and (84) back into (82), and using (88) and (92), we have that

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) \leq \mathcal{F}(\theta_{t+1}, \mu_t) - \gamma \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \quad (93)$$

$$+ (M + \alpha^2) B^2 \int_0^\gamma (\gamma - \tau) \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 d\tau \quad (94)$$

$$\leq \mathcal{F}(\theta_{t+1}, \mu_t) - \gamma \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2 \quad (95)$$

$$+ \left(\frac{M + \alpha^2}{2} \right) B^2 \gamma^2 \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2, \quad (96)$$

and thus, in particular, that

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_{t+1}, \mu_t) \leq -\gamma \left(1 - \frac{(M + \alpha^2) B^2 \gamma}{2} \right) \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2. \quad (97)$$

In order to apply these results to obtain (97), we need to show that the sequence $(\xi_t)_{t \in \mathbb{N}}$ satisfies the required condition on the step size in (79). In particular, we need to establish that, for each $t \in \mathbb{N}$, under the assumption that $\gamma < \gamma_*$, it holds that

$$\gamma \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{d_z}} \leq \frac{\alpha - 1}{\alpha B}. \quad (98)$$

In fact, we will establish this bound by proving that, under our assumptions, a slightly stronger result holds, namely, that for all $\theta \in \tilde{\Theta}$, where $\tilde{\Theta} = \{\theta_t\}_{t \in \mathbb{N}}$, it holds that

$$\gamma \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta, \mu_t)\|_{\mathcal{H}_k^{d_z}} \leq \frac{\alpha - 1}{\alpha B}. \quad (99)$$

Similar to the proof of (Salim et al., 2022, Theorem 3.2), we prove this via induction. First note, arguing similarly to the proof of (Salim et al., 2022, Lemma C.1), that for each $\theta \in \Theta$,

$$\|\mathcal{S}_{\mu} \nabla_{W_2} \mathcal{F}(\theta, \mu)\|_{\mathcal{H}_k^{d_z}} = \|\mathbb{E}_{z \sim \mu} [\nabla_z \log \pi_{\theta}(z) k(z, \cdot) + \nabla_z k(z, \cdot)]\|_{\mathcal{H}_k^{d_z}} \quad (100)$$

$$\leq \mathbb{E}_{z \sim \mu} \|\nabla_z \log \pi_{\theta}(z) k(z, \cdot)\|_{\mathcal{H}_k^{d_z}} + \mathbb{E}_{z \sim \mu} \|\nabla_z k(z, \cdot)\|_{\mathcal{H}_k^{d_z}} \quad (101)$$

$$\leq B [1 + \|\nabla_z \log p_{\theta}(0|x)\|] + M \mathbb{E}_{z \sim \mu} [\|z\|] \quad (102)$$

$$\leq B [1 + \|\nabla_z \log p_{\theta}(0|x)\|] + M [W_1(p_{\theta}(\cdot|x), \delta_0) + W_1(\mu, p_{\theta}(\cdot|x))] \quad (103)$$

$$\leq B \left[1 + \|\nabla_z \log p_{\theta}(0|x)\| + M \int_{\mathcal{Z}} \|z\| p_{\theta}(z|x) dz + M \sqrt{\frac{2\text{KL}(\mu \| p_{\theta}(\cdot|x))}{\lambda}} \right] \quad (104)$$

$$= BC_{\theta} + BM \sqrt{\frac{2\text{KL}(\mu \| p_{\theta}(\cdot|x))}{\lambda}} \quad (105)$$

where (101) follows from the triangle inequality, (102) follows from Assumption 1 and Assumption 2, (103) follows from the triangle inequality for the metric W_1 , (104) follows from Assumption 3, and in the final line we have defined

$$C_{\theta} = 1 + \|\nabla_z \log p_{\theta}(0|x)\| + M \int_{\mathcal{Z}} \|z\| p_{\theta}(z|x) dz. \quad (106)$$

Now, suppose that $0 < \gamma < \gamma^*$. Then, using the definition of γ^* given in Theorem 1, and the bound in (105), it follows straightforwardly that, for any $\theta \in \tilde{\Theta}$,

$$\gamma \|\mathcal{S}_{\mu_0} \nabla_{W_2} \mathcal{F}(\theta, \mu_0)\|_{\mathcal{H}_k^{d_z}} \leq \gamma \left[BC_{\theta} + BM \sqrt{\frac{2\text{KL}(\mu_0 \| p_{\theta}(\cdot|x))}{\lambda}} \right] \quad (107)$$

$$\leq \frac{\alpha - 1}{\alpha B^2} \left[C_{\theta} + M \sqrt{\frac{2\text{KL}(\mu_0 \| p_{\theta}(\cdot|x))}{\lambda}} \right]^{-1} \left[BC_{\theta} + BM \sqrt{\frac{2\text{KL}(\mu_0 \| p_{\theta}(\cdot|x))}{\lambda}} \right] \quad (108)$$

$$\leq \frac{\alpha - 1}{\alpha B}. \quad (109)$$

This establishes that (99) holds for $t = 0$. We now prove the inductive step. Suppose that, for each $t \in [T - 1]$, $0 < \gamma \leq \gamma^*$ implies that, for each $\theta \in \tilde{\Theta}$,

$$\gamma \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta, \mu_t)\|_{\mathcal{H}_k^{d_z}} \leq \frac{\alpha - 1}{\alpha B}. \quad (110)$$

Using this assumption, we can substitute (83) and (84) into (82), and use (88) and (92), but now with $\xi = P_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta, \mu_t)$. This implies that, for all $t \in [T - 1]$, and for all $\theta \in \tilde{\Theta}$,

$$\mathcal{F}(\theta, \mu_{t+1}) - \mathcal{F}(\theta, \mu_t) \leq -\gamma \left(1 - \frac{(M + \alpha^2)B^2\gamma}{2} \right) \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta, \mu_t)\|_{\mathcal{H}_k^{d_z}}^2. \quad (111)$$

Thus, in particular, $\mathcal{F}(\theta, \mu_T) \leq \mathcal{F}(\theta, \mu_0)$ for all $\theta \in \tilde{\Theta}$. It follows straightforwardly that

$$BC_{\theta} + BM \sqrt{\frac{2\text{KL}(\mu_T \| p_{\theta}(\cdot|x))}{\lambda}} = BC_{\theta} + BM \sqrt{\frac{2[\mathcal{F}(\theta, \mu_T) + \log p_{\theta}(x)]}{\lambda}} \quad (112)$$

$$\leq BC_{\theta} + BM \sqrt{\frac{2[\mathcal{F}(\theta, \mu_0) + \log p_{\theta}(x)]}{\lambda}} = BC_{\theta} + BM \sqrt{\frac{2\text{KL}(\mu_0 \| p_{\theta}(\cdot|x))}{\lambda}}. \quad (113)$$

By combining this display with the bound in (105), and arguing as in (109), we then have the required result in (99). In particular, for all $\theta \in \tilde{\Theta}$, we have

$$\gamma \|\mathcal{S}_{\mu_T} \nabla_{W_2} \mathcal{F}(\theta, \mu_T)\|_{\mathcal{H}_k^{dz}} \leq \gamma \left[BC_\theta + BM \sqrt{\frac{2\text{KL}(\mu_T \| p_\theta(\cdot|x))}{\lambda}} \right] \leq \gamma \left[BC_\theta + BM \sqrt{\frac{2\text{KL}(\mu_0 \| p_\theta(\cdot|x))}{\lambda}} \right] \leq \frac{\alpha - 1}{\alpha B}. \quad (114)$$

This completes the proof of (99), which in turn implies (98), and thus (97). Finally, combining (76) and (97), we have the desired bound,

$$\mathcal{F}(\theta_{t+1}, \mu_{t+1}) - \mathcal{F}(\theta_t, \mu_t) \leq -\gamma \left[\left(1 - \frac{M\gamma}{2}\right) \|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \left(1 - \frac{(M + \alpha^2)B^2\gamma}{2}\right) \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{dz}}^2 \right]. \quad (115)$$

This completes the proof of Theorem 1, with constants

$$A_1 = 1 - \frac{M\gamma}{2}, \quad A_2 = 1 - \frac{(M + \alpha^2)B^2\gamma}{2}. \quad (116)$$

□

B.3 Proof of Corollary 1.

Proof. Using Theorem 1, iterating, and using the definition of c_γ given in Corollary 1, we have that

$$\mathcal{F}(\theta_{T+1}, \mu_{T+1}) - \mathcal{F}(\theta_0, \mu_0) \leq -c_\gamma \sum_{t=0}^T \left(\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{dz}}^2 \right), \quad (117)$$

Thus, rearranging, and using the fact that $\min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu) \leq \mathcal{F}(\theta_{T+1}, \mu_{T+1})$, we have as required that

$$\frac{1}{T} \sum_{t=0}^T \left(\|\nabla_\theta \mathcal{F}(\theta_t, \mu_t)\|_{\mathbb{R}^{d_\theta}}^2 + \|\mathcal{S}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_{t+1}, \mu_t)\|_{\mathcal{H}_k^{dz}}^2 \right) \leq \frac{\mathcal{F}(\theta_0, \mu_0) - \min_{(\theta, \mu) \in \Theta \times \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu)}{c_\gamma T}. \quad (118)$$

□

B.4 Proof of Theorem 2

Proof. Under Assumption 4, the marginal likelihood $\theta \mapsto p_\theta(x)$ admits a unique maximizer $\theta^* = \arg \max_{\theta \in \Theta} p_\theta(x)$ (Kuntz et al., 2023, Theorem 4). Thus, in particular, the free energy $(\theta, \mu) \mapsto \mathcal{F}(\theta, \mu)$ admits a unique minimizer (θ^*, μ^*) , with $\mu^* = p_{\theta^*}(\cdot|x)$ (Kuntz et al., 2023, Theorem 1).

Next, once more appealing to Assumption 4, the function $(\theta, \mu) \mapsto \mathcal{F}(\theta, \mu)$ is (geodesically) convex on the product space $\Theta \times \mathcal{P}_2(\mathcal{Z})$. In particular, convexity w.r.t. the first argument follows directly from the definition of the free-energy, while geodesic convexity w.r.t. the second argument follows from (Ambrosio et al., 2008, Proposition 9.3.2, Theorem 9.4.12). We thus have, using Jensen's inequality for (119) and the first-order characterization of a (geodesically) convex function (e.g., Ambrosio et al., 2008, Chapter 10) for (120), that

$$\mathcal{F} \left(\frac{1}{T} \sum_{t=1}^T \theta_t, \frac{1}{T} \sum_{t=1}^T \mu_t \right) - \mathcal{F}(\theta^*, \mu^*) \leq \frac{1}{T} \sum_{t=1}^T [\mathcal{F}(\theta_t, \mu_t) - \mathcal{F}(\theta^*, \mu^*)] \quad (119)$$

$$\begin{aligned} &\leq \frac{1}{T} \sum_{t=1}^T \langle -\nabla_\theta \mathcal{F}(\theta_t, \mu_t), \theta_t - \theta^* \rangle_{\mathbb{R}^{d_\theta}} \\ &\quad + \frac{1}{T} \sum_{t=1}^T \int_{\mathbb{R}^{dz}} \langle -\nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)(z), t_{\mu_t}^{\mu^*}(z) - z \rangle_{\mathbb{R}^{dz}} \mu_t(z) dz. \end{aligned} \quad (120)$$

The required bound given in Theorem 2 (see App. A.3) now follows from existing results, arguing as in the proof of (Orabona and Pal, 2016a, Corollary 5) with $\varepsilon = 1$ to bound the first term,⁵ and arguing as in the proof of (Sharrock and Nemeth, 2023, Proposition 3.3') with $w_0 = 1$ and replacing $\nabla_{W_2} \mathcal{F}(\mu_t) \mapsto \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t)$ to bound the second term. □

⁵For an informal overview of this approach, see also Part II of Orabona and Cutkosky (2020).

C MARGINAL SVGD EM AND MARGINAL COIN EM

In certain models, the M step is tractable. In other words, it is possible to derive a tractable expression for $\theta_*(\mu) = \arg \max_{\mu \in \mathcal{P}_2(\mathcal{Z})} \mathcal{F}(\theta, \mu)$. Thus, in particular, given the empirical measure $\mu^N = \frac{1}{N} \sum_{j=1}^N \delta_{z^j}$, we can compute $\theta_*(z^{1:N}) := \theta_*(\mu^N)$, where $z^{1:N} = (z^1, \dots, z^N) \in \mathcal{Z}^N$. In such cases, instead of SVGD EM (Alg. 1) or Coin EM (Alg. 2), we can use marginal variants of these algorithms, in which the θ updates are now exact.

Algorithm 4 Marginal SVGD EM

- 1: **input:** number of iterations T , number of particles N , initial particles $\{z_0^i\}_{i=1}^N \sim \mu_0$, initial θ_0 , target density π , kernel k , function $\mu \mapsto \theta_*(\mu)$, learning rate γ .
- 2: **for** $t = 1, \dots, T - 1$ **do**
- 3: For $i \in [N]$

$$z_{n+1}^i = z_n^i + \frac{\gamma}{N} \sum_{j=1}^N \left[k(z_t^j, z_t^i) \nabla_z \log \pi_{\theta_*(z_t^{1:N})}(z_t^j) + \nabla_{z_t^j} k(z_t^j, z_t^i) \right]$$

- 4: **return** θ_T and $\{z_T^i\}_{i=1}^N$.
-

Algorithm 5 Marginal Coin EM

- 1: **input:** number of iterations T , number of particles N , initial particles $\{z_0^i\}_{i=1}^N \sim \mu_0$, initial θ_0 , target density π , kernel k , function $\mu \mapsto \theta_*(\mu)$.
- 2: **for** $t = 1, \dots, T$ **do**
- 3: For $i \in [N]$

$$z_t^i = z_0^i + \frac{\sum_{s=1}^{t-1} \frac{1}{N} \sum_{j=1}^N k(z_s^j, z_s^i) \nabla_z \log \pi_{\theta_*(z_s^{1:N})}(z_s^j) + \nabla_{z_s^j} k(z_s^j, z_s^i)}{t} \\ \times \left(1 + \sum_{s=1}^{t-1} \left\langle \frac{1}{N} \sum_{j=1}^N k(z_s^j, z_s^i) \nabla_z \log \pi_{\theta_*(z_s^{1:N})}(z_s^j) + \nabla_{z_s^j} k(z_s^j, z_s^i), z_s^i - z_0^i \right\rangle \right)$$

- 4: **return** θ_T and $\{z_T^i\}_{i=1}^N$.
-

D ADAPTIVE COIN EM

In Sec. 2.3, the update equation given in (15), and thus the update equations in Alg. 2, were given under the assumption that the sequence of outcomes, in this case $(c_t^\theta)_{t \in [T]} = (\nabla_{\theta} \mathcal{F}(\theta_t, \mu_t^N))_{t \in [T]}$ and $(c_t^\mu)_{t \in [T]} = (\mathcal{P}_{\mu_t} \nabla_{W_2} \mathcal{F}(\theta_t, \mu_t^N))_{t \in [T]}$, were bounded above by 1 (see the remark in Footnote 2). In practice, of course, this may not be the case. If, instead, there exists a known constant which upper bounds $(c_t^\theta)_{t \in [T]}$ and $(c_t^\mu)_{t \in [T]}$, then one can simply replace these quantities by their normalized versions in Alg. 2. In the more unlikely scenario that such a constant is unknown, we can instead use an adaptive version of Coin EM, in which an empirical estimate of this constant is updated as the algorithm progresses; see (Orabona and Tommasi, 2017, Sec. 6) and (Sharrock and Nemeth, 2023, App. D) for precedents. This algorithm, which we recommend and use in all experiments, is summarized in Alg. 6.

In addition, once more following (Orabona and Tommasi, 2017, Sec. 6), whenever we apply Coin EM to a Bayesian neural network (see Sec. 4.3 and App. E.4), we further alter the parameter update equation in Alg. 6 to read as

$$\theta_{t,j} = \theta_{0,j} + \frac{\sum_{s=1}^{t-1} c_{s,j}^\theta}{\max(G_{t,j}^\theta + L_{t,j}^\theta, 100L_{t,j}^\theta)} \left(1 + \frac{R_{t,j}^\theta}{L_{t,j}^\theta} \right), \quad (121)$$

with an analogous modification for the particle update equation. It is worth noting that both of these adaptive versions of the Coin EM algorithm remain entirely tuning free.

Algorithm 6 Adaptive Coin EM

- 1: **input:** number of iterations T , number of particles N , initial particles $\{z_0^i\}_{i=1}^N \sim \mu_0$, initial θ_0 , target density π , kernel k
- 2: **initialize:** for $i \in [N]$: $L_{0,j}^\theta = 0$, $G_{0,j}^\theta = 0$, $R_{0,j}^\theta = 0$; for $i \in [N]$ and $j \in [d]$, $L_{0,j}^{z,i} = 0$, $G_{0,j}^{z,i} = 0$, $R_{0,j}^{z,i} = 0$.
- 3: **for** $t = 1, \dots, T$ **do**
- 4: Compute

$$c_{t-1}^\theta = \frac{1}{N} \sum_{j=1}^N \nabla_\theta \log \pi_{\theta_{t-1}}(z_{t-1}^j) \quad (\text{parameter gradient})$$

- 5: **for** $j = 1, \dots, d_\theta$ **do**
- 6: Compute

$$L_{t,j}^\theta = \max(L_{t-1,j}^\theta, |c_{t,j}^\theta|) \quad (\text{max. observed scale})$$

$$G_{t,j}^\theta = G_{t-1,j}^\theta + |c_{t-1,j}^\theta| \quad (\text{sum of absolute value of gradients})$$

$$R_{t,j}^\theta = \max(R_{t-1,j}^\theta + \langle c_{t-1,j}^\theta, \theta_{t-1,j} - \theta_{0,j} \rangle, 0) \quad (\text{total reward})$$

$$\theta_{t,j} = \theta_{0,j} + \frac{\sum_{s=1}^{t-1} c_{s,j}^\theta}{G_{t,j}^\theta + L_{t,j}^\theta} \left(1 + \frac{R_{t,j}^\theta}{L_{t,j}^\theta}\right). \quad (\text{parameter update})$$

- 7: **for** $i = 1, \dots, N$ **do**
- 8: Compute

$$c_{t-1}^{z,i} = \frac{1}{N} \sum_{j=1}^N k(z_{t-1}^j, z_{t-1}^i) \nabla_{z^j} \log \pi_{\theta_t}(z_{t-1}^j) + \nabla_{z^j} k(z_{t-1}^j, z_{t-1}^i) \quad (\text{particles gradient})$$

- 9: **for** $j = 1, \dots, d_z$ **do**
- 10: Compute

$$L_{t,j}^{z,i} = \max(L_{t-1,j}^{z,i}, |c_{t-1,j}^{z,i}|) \quad (\text{max. observed scale})$$

$$G_{t,j}^{z,i} = G_{t-1,j}^{z,i} + |c_{t-1,j}^{z,i}| \quad (\text{sum of absolute value of gradients})$$

$$R_{t,j}^{z,i} = \max(R_{t-1,j}^{z,i} + \langle c_{t-1,j}^{z,i}, z_{t-1,j}^i - z_{0,j}^i \rangle, 0) \quad (\text{total reward})$$

$$z_{t,j}^i = z_{0,j}^i + \frac{\sum_{s=1}^{t-1} c_{s,j}^{z,i}}{G_{t,j}^{z,i} + L_{t,j}^{z,i}} \left(1 + \frac{R_{t,j}^{z,i}}{L_{t,j}^{z,i}}\right). \quad (\text{particles update})$$

- 11: **output:** θ_T and $(z_T^i)_{i=1}^N$.
-

E ADDITIONAL EXPERIMENTAL DETAILS

We implement all of the algorithms using Python 3 and JAX (Bradbury et al., 2018). We perform all experiments using a MacBook Pro 16" (2021) laptop with Apple M1 Pro chip and 16GB of RAM.

E.1 Toy hierarchical model

Implementation Details. For the results in Fig. 1, we initialize all methods with $\theta_0 \sim \mathcal{N}(0, 0.1^2)$ and $z_0^i \sim \mathcal{N}(0, 1)$. We use $N = 10$ particles and run each algorithm for $T = 500$ iterations. Additional results for larger numbers of particles are given in App. F.1. In Fig. 1(a), we run each algorithm over a grid of 50 logarithmically spaced learning rates $\gamma \in [10^{-5}, 10^3]$. We use Adagrad Duchi et al. (2011) to automatically adjust the learning rates of SVGD EM and PGD. In Fig. 1(b) and Fig. 1(c), we run each algorithm using the learning rate which obtained the lowest final iterate MSE in Fig. 1(a).

For Fig. 2(a) and Fig. 2(b), we run the algorithms for $T = 5000$ iterations, and report the empirical variance of the latent particles $(z_t^i)_{i=1}^N$ for each $t \in [0, T]$. In Fig. 2(c), we run each algorithm for $T = 250$ iterations with

$N = 50$ particles. For Coin EM and SVGD EM, we compute the MSE between the empirical variance of the final particles $(z_T^i)_{i=1}^N$, and the true posterior variance, which in this case is just given by 0.5 (Kuntz et al., 2023, App. E.1). For PGD and SOUL, we use the empirical variance of time-averaged particles, due to the noise used in the particle updates. All results are averaged over 10 random seeds (5 random seeds for Fig. 2(c)), and we report 95% bootstrap CIs as well as the mean.

E.2 Bayesian logistic regression

Model. We consider the setup in (De Bortoli et al., 2021, Sec. 4.1). The observed data is $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^{d_x}$ are d_x -dimensional real-valued covariates, and $y_i \in \{0, 1\}$ are binary class labels. We assume the labels y_i are conditionally independent given the features x_i and the regression weights $z \in \mathbb{R}^{d_z}$, with $p(\{y_i, x_i\}|z) = \sigma(x_i^T z)^{y_i} [1 - \sigma(x_i^T z)]^{1-y_i}$ for $i \in [N]$, where $\sigma(u) := e^u / (1 + e^u)$ denotes the standard logistic function. We place a Gaussian prior on the latent weights, $p(z) = \mathcal{N}(z | \theta \mathbf{1}_{d_z}, 5 \mathbf{1}_{d_z})$, for some real parameter $\theta \in \mathbb{R}$ which we would like to estimate. The model is thus

$$p_\theta(z, \mathcal{D}) = \mathcal{N}(z | \theta \mathbf{1}_{d_z}, 5 \mathbf{1}_{d_z}) \prod_{i=1}^N \sigma(x_i^T z)^{y_i} [1 - \sigma(x_i^T z)]^{1-y_i}. \quad (122)$$

Data. We fit this model using the Wisconsin Breast Cancer dataset Wolberg and Mangasarian (1990). This dataset contains 683 datapoints, each with nine features $x_i \in \mathbb{R}^9$ extracted from a digitized image of a fine needle aspirate of a breast mass, and a label $y_i \in \{0, 1\}$ indicating whether the mass is benign ($y_i = 0$) or malign ($y_i = 1$). We normalize the features so that each has mean zero and unit standard deviation across the dataset, and use a random 80-20 train-test split.

Implementation Details. For the results in Fig. 3, following De Bortoli et al. (2021); Kuntz et al. (2023), we initialize $\theta_0 = 0$ and $z_0^i \sim \mathcal{N}(0, 1)$. In Fig. 3(a) and Fig. 3(b), we use a learning rate of $\gamma = 0.02$ for PGD, PMGD, and SOUL, and a learning rate of $\gamma = 0.2$ for SVGD EM, which typically obtains its best performance for a higher learning rate than the other methods (see, e.g., Fig. 1(a) or Fig. 2(c)). We run each algorithm with $N = 100$ particles for $T = 800$ iterations. In Fig. 3(c), we run each algorithm using a grid of 50 logarithmically spaced learning rates $\gamma \in [10^{-10}, 10^2]$, using $N = 20$ particles and $T = 400$ iterations. We repeat each experiment over 10 random test-train splits of the data.

E.3 Bayesian logistic regression (alternative model)

In App. F.3, we present results for an alternative Bayesian logistic regression model, which were omitted from the main text due to space constraints. Here, we provide full details of this model.

Model. We now follow the setup in Gershman et al. (2012). Similar to before, the observed data is given by $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^{d_x}$ are d_x -dimensional real-valued covariates, and $y_i \in \{0, 1\}$ are binary class labels. Meanwhile, the latent variables are now given by $z = \{w, \log \alpha\}$, consisting of a d_x -dimensional real-valued regression coefficients $w_k \in \mathbb{R}^{d_x}$, and a positive precision parameter $\alpha \in \mathbb{R}_+$. Finally, the parameters are given by $\theta = (\log a, \log b)$, for positive $a, b \in \mathbb{R}_+$.

We assume that the labels y_i are conditionally independent given the features x_i and the regression weights $z \in \mathbb{R}^{d_z}$, with $p(\{y_i, x_i\}|z) = \sigma(x_i^T z)^{y_i} [1 - \sigma(x_i^T z)]^{1-y_i}$ for $i \in [N]$, where $\sigma(u) := e^u / (1 + e^u)$ denotes the standard logistic function. We now also assume that $p(w_k | a) = \mathcal{N}(w_k; 0, \alpha^{-1})$ for $k \in [d_x]$, and place a Gamma prior on α , so $p(\alpha) = \Gamma(\alpha | a, b)$. In this case, the model is given by

$$p_\theta(z, x) = \text{Gamma}(\alpha | a, b) \prod_{k=1}^{d_x} \mathcal{N}(w_k; 0, \alpha^{-1}) \prod_{i=1}^N \sigma(x_i^T z)^{y_i} [1 - \sigma(x_i^T z)]^{1-y_i}. \quad (123)$$

Data. We fit this model to several datasets from the UCI Machine Learning repository Dua and Graff (2019): the Covtype dataset, which contains 581012 datapoints and 54 attributes, the Banknote dataset, which contains 1372 datapoints and 4 features, and the Cleveland heart disease dataset, which contains 303 datapoints and 13 features. We split the dataset into train-test sets using a 70-30 train-test split.

Implementation Details. For the results in Fig. 13, we initialize $a_0 = 1$, $b_0 = 0.01$, $\alpha_0^i \sim \text{Gamma}(a_0, b_0)$, and $w_0^i \sim \mathcal{N}(0, 1/\alpha_0^i)$. We run each algorithm using $N = 10$ particles for $T = 1000$ iterations, over a grid of

logarithmically spaced learning rates $\gamma \in [10^{-9}, 10^1]$. We repeat each experiment over 10 random train-test splits of the data.

E.4 Bayesian neural network

Model. We consider the setting described in (Yao et al., 2022, Section 6.5) and (Kuntz et al., 2023, Sec. 3.2), applying a neural network to classify MNIST images (Lecun et al., 1998). In particular, we use a Bayesian two-layer neural network with tanh activation functions, a softmax output layer, and Gaussian priors on the weights.

In this case, the observed data is of the form $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^{784}$ are 28×28 grayscale images of handwritten digits, and $y_i \in \{0, 1\}$ are labels denoting whether the image is a 4 or 9. We normalise the 784 features so that each has mean zero and unit standard deviation across the dataset. We assume the labels y_i are conditionally independent given the features x_i and the network weights $z = (w, v)$, where $w \in \mathbb{R}^{d_w=40 \times 784}$ and $v \in \mathbb{R}^{d_v=2 \times 40}$, and that

$$p(y_i|x_i, w) = \exp\left(\sum_{j=1}^{40} v_{ij} \tanh\left(\sum_{k=1}^{784} w_{jk} x_{ik}\right)\right) \quad (124)$$

for $i \in [n]$. We then place Gaussian priors on the weights of input layer and the output layer, viz $p(w_k) = \mathcal{N}(w_k|0, e^{2\alpha})$, for $k \in [784]$ and $p(v_j) = \mathcal{N}(v_j|0, e^{2\beta})$ for $j \in [40]$. In this case, rather than assigning a hyperprior to α and β , we will learn them from data. Thus, our model takes the form

$$p_\theta(z, \mathcal{D}) = \prod_{j=1}^{40} \mathcal{N}(w_k|0, e^{2\alpha}) \prod_{k=1}^{784} \mathcal{N}(v_k|0, e^{2\beta}) \prod_{i=1}^N p(y_i|x_i, w) \quad (125)$$

where $z = (w, v) \in \mathbb{R}^{40 \times 784 + 2 \times 40}$, and $\theta = (\alpha, \beta)$.

Data. We use the MNIST dataset Lecun et al. (1998), which contains 70'000 28×28 grayscale images $x_i \in \mathbb{R}^{784}$ of handwritten digits between 0 and 9, each with an accompanying label $y_i \in \{0, 1, \dots, 9\}$. Following Kuntz et al. (2023); Yao et al. (2022), we subsample 1000 points with labels 4 and 9. We normalize the features so that each has mean zero and unit standard deviation across the dataset, and split the data using a random 80-20 train-test split.

Implementation Details. For the results in Fig. 14 - 16, we initialize $\alpha_0 = 0$, $\beta_0 = 0$, and $w_k^i \sim \mathcal{N}(0, e^{2\alpha_0})$, $v_j \sim \mathcal{N}(0, e^{2\beta_0})$, and run each algorithm for $T = 500$ iterations. In addition, in Fig. 16, we use $N = 10$ particles, and run each algorithm for 50 logarithmically spaced learning rates $\gamma \in [10^{-9}, 10^1]$. We repeat each experiment over 5 (Fig. 14 - 4) or 10 (Fig. 16) random train-test splits of the data.

E.5 Bayesian neural network (alternative model)

In App. F.5, we present results for an alternative Bayesian neural network model, which were omitted from the main text due to space constraints. Here, we provide full details of this model.

Model. We consider the setup described in Hernandez-Lobato and Adams (2015); Liu and Wang (2016). The observed data is of the form $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^{d_x}$ are d_x -dimensional real-valued covariates, and $y_i \in \mathbb{R}$ are real-valued responses. We assume that the responses y_i are conditionally independent given the covariates x_i and the network weights $w \in \mathbb{R}^{d_w}$, and that $p(y_i|x_i, w) = \mathcal{N}(y_i|f(x_i, w), \gamma^{-1})$ for $i \in [n]$, where $f(x_i, w)$ denotes the output of the neural network. We place a Gaussian prior on each of the neural network weights, namely $p(w_j) = \mathcal{N}(w_j|0, \lambda^{-1})$, for $j \in [d_w]$. To complete our model, we assume a Gamma prior for the inverse covariance $\gamma \in \mathbb{R}_+$, and a Gamma hyperprior on the inverse covariance $\lambda \in \mathbb{R}_+$, so $p(\gamma) = \text{Gamma}(\gamma|a_\gamma, b_\gamma)$ $p(\lambda) = \text{Gamma}(\lambda|a_\lambda, b_\lambda)$. Our model thus takes the form

$$p_\theta(z, \mathcal{D}) = \text{Gamma}(\gamma|a_\gamma, b_\gamma) \text{Gamma}(\lambda|a_\lambda, b_\lambda) \prod_{i=1}^N \prod_{j=1}^{d_z} \mathcal{N}(y_i|f(x_i, w), \gamma^{-1}) \mathcal{N}(w_j|0, \lambda^{-1}) \quad (126)$$

where $z = (w, \log \lambda, \log \gamma) \in \mathbb{R}^{d_w+1+1}$, and $\theta = (\log a_\gamma, \log b_\gamma, \log a_\lambda, \log b_\lambda)$. Rather than fixing the parameters as in Hernandez-Lobato and Adams (2015); Liu and Wang (2016), we allow these parameters to be learned from the data.

Data. We fit the Bayesian neural network using several UCI datasets [Dua and Graff \(2019\)](#). The number of datapoints varies from 506 in the Boston Housing (*Boston*) dataset, to 11934 in the Naval Propulsion (*Naval*) dataset. The number of features is between 4 in the Combined Cycle Power Plant (*Power*) to 16 in the Naval Propulsion dataset (*Naval*). In each case, we normalize the features so that each has mean zero and unit standard deviation across the dataset, and split the data using a random 90-10 train-test split.

Implementation Details. For the results in Fig. 17, we use a neural network with a single hidden layer and with 50 hidden units, and a Relu activation function. We use Adagrad [Duchi et al. \(2011\)](#) to automatically adjust the learning rates of SVGD EM and PGD. We initialize the parameters by setting $a_{\gamma_0} = 1$, $b_{\gamma_0} = 0.1$, $a_{\lambda_0} = 1$, $b_{\lambda_0} = 0.1$. Meanwhile, we initialize the particles for the precisions as $\lambda \sim \text{Gamma}(a_{\lambda_0}, b_{\lambda_0})$ and $\gamma \sim \text{Gamma}(a_{\gamma_0}, b_{\gamma_0})$, and for the weights w according to zero-mean Gaussians with variance given by the reciprocal of the input dimension (the latent dimension for the first layer, and the number of hidden units for the second). We run each algorithm using $N = 20$ particles for $T = 1000$ iterations, over a logarithmically spaced grid of 30 learning rates $\gamma \in [10^{-9}, 10^1]$. We repeat each experiment over 10 random train-test splits of the data.

E.6 Latent space network model

Model. We consider the latent space network model first introduced in [Hoff et al. \(2002\)](#), where, in the context of social networks, each entity is represented by a node of the network. Edges between nodes indicate connections between entities. We restricted ourselves to binary undirected graphs, where an edge in the network between nodes i and j is represented by a binary indicator, $y_{i,j} = \{0, 1\}$.

The latent space network model [Hoff et al. \(2002\)](#); [Turnbull et al. \(2019\)](#) is popular network model to simplify complex network data by embedding the network’s nodes in a lower-dimensional space z . A general form of this model is,

$$\begin{aligned} Y_{i,j} &\sim \text{Bernoulli}(p_{ij}) \quad i \neq j; \quad i, j = 1, \dots, n \\ \text{logit}(p_{ij}) &= \theta + d(z_i, z_j), \end{aligned} \tag{127}$$

where $d(\cdot, \cdot)$ is a function of the similarity between the latent variables. In this paper, we let $d(z_i, z_j) = \|z_i - z_j\|$ be the Euclidean distance between z_i and z_j , although other similarity metrics which preserve the triangle inequality are also possible.

Data. We use a dataset curated by [Beveridge and Chemers \(2018\)](#) from fan scripts on the website Genius. The dataset contains the frequency of interactions among characters in the first four season of the TV series Game of Thrones. Each season is represented by a binary undirected network, where an edge indicates that two characters interacted at least 10 times in that season. For the purpose of visualization, we filter out minor characters with fewer than 10 interactions per season. The dataset covers four seasons of the TV series, one network per season, with $n = 165$ nodes each season.

Implementation Details. We follow a similar inference scheme as [Hoff et al. \(2002\)](#). Firstly, we find the maximum likelihood estimates (MLEs) for (θ, z) by maximizing the likelihood function (127). Using the MLEs $(\hat{\theta}, \hat{z})$, we initialize PGD, SOUL and Coin EM with $\theta_0 = \hat{\theta}$ and for $i = 1, \dots, N$, set $z_0^i \sim \mathcal{N}(\hat{z}, 0.1)$. One of the well-known challenges of using latent space network models is that a set of points in Euclidean space will have the same distances regardless of how they are rotated, reflected, or translated. This means that the likelihood (127) is invariant to transformation of the latent positions z . This issue can be resolved by using a Procrustes transformation of the latent variables, relative to a fixed point, which we choose to be the MLE, \hat{z} . Therefore, for all z_t^i in all algorithms we store and plot $\tilde{z}_t^i = \arg\min_{Tz} \text{tr}(\hat{z} - Tz_t^i)^\top (\hat{z} - Tz_t^i)$. For the SOUL and PGD algorithms we use learning rate parameter $\gamma = 0.001$. For all algorithms we use $N = 10$ particles at $T = 500$ iterations.

F ADDITIONAL NUMERICAL RESULTS

F.1 Toy hierarchical model

In this section, we provide additional results for the toy hierarchical model studied in Sec. 4.1.

Additional results for a different number of particles. We first consider the impact of varying the number of particles. In Fig. 6 and Fig. 7, we replicate the results in Fig. 1, now using $N = 20$ and $N = 50$ particles,

rather than $N = 10$. There is little to distinguish between the two sets of results. In particular, all methods still converge to θ_* ; and Coin EM and SVGD EM still tend to achieve a lower MSE than PGD and SOUL. We note that the appearance of wide blue bands in Fig. 6 and Fig. 7 (and Fig. 1) is an artifact of the plotting resolution, and the logarithmic scale. In truth, the Coin EM parameter estimates oscillate rapidly as they convergence towards the marginal maximum likelihood estimate, giving the appearance of these bands at this resolution. This type of oscillatory convergence is common for coin betting algorithms; see, e.g., (Orabona and Cutkosky, 2020, pg. 36) for a very simple example.

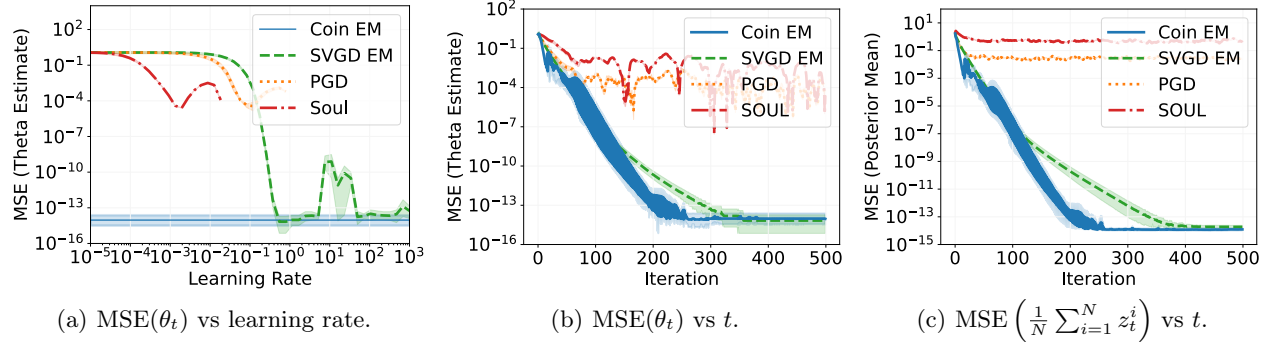


Figure 6: **Additional results for the toy hierarchical model with $N = 20$ particles.** MSE of the parameter estimate θ_t as a function of the learning rate after $T = 500$ iterations (a); and MSE of the parameter estimate (b) and the posterior mean (c) as a function of the number of iterations, using the optimal learning rate from (a).

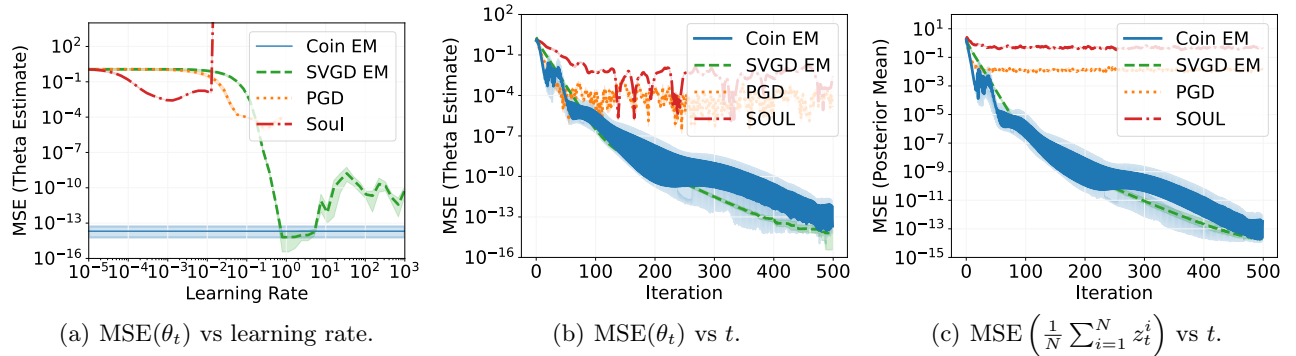


Figure 7: **Additional results for the toy hierarchical model with $N = 50$ particles.** MSE of the parameter estimate θ_t as a function of the learning rate after $T = 500$ iterations (a); and MSE of the parameter estimate (b) and the posterior mean (c) as a function of the number of iterations, using the optimal learning rate from (a).

Additional results for different learning rates. Next, we provide an additional demonstration of how the choice of learning rate can affect the parameter estimates generated by PGD, SOUL, and SVGD EM. In particular, in Fig. 8, we plot the parameter estimates generated by SVGD EM, PGD, and SOUL for three different learning rates: the optimal learning rate as determined by Fig. 7(a), a smaller learning rate, and a larger learning rate ‘at the edge of stability’. The specific learning rates $\{\gamma_{\text{opt}}, \gamma_{\text{small}}, \gamma_{\text{large}}\}$ used in this figure $\{0.79, 0.039, 100.00\}$ for SVGD EM,⁶ $\{0.26, 0.013, 1.15\}$ for PGD, and $\{0.0013, 0.000066, 0.018\}$ for SOUL.

Additional results for a different initialization. We now consider the impact of varying the initialization. In Fig. 9, we repeat the experiment in Sec. 4.1, but now using an initialization far away from the true parameter $\theta = 1$. In particular, we now initialize the parameters using $\theta_0 \sim \mathcal{N}(10, 0.1^2)$. The results are rather similar to before. In particular, Coin EM and SVGD EM both converge rapidly to the true parameter, and tend to outperform the competitors.

⁶We note that, when SVGD EM is implemented with an adaptive method such as Adagrad Duchi et al. (2011), as is the case here, it remains stable even for very large values of the learning rate.

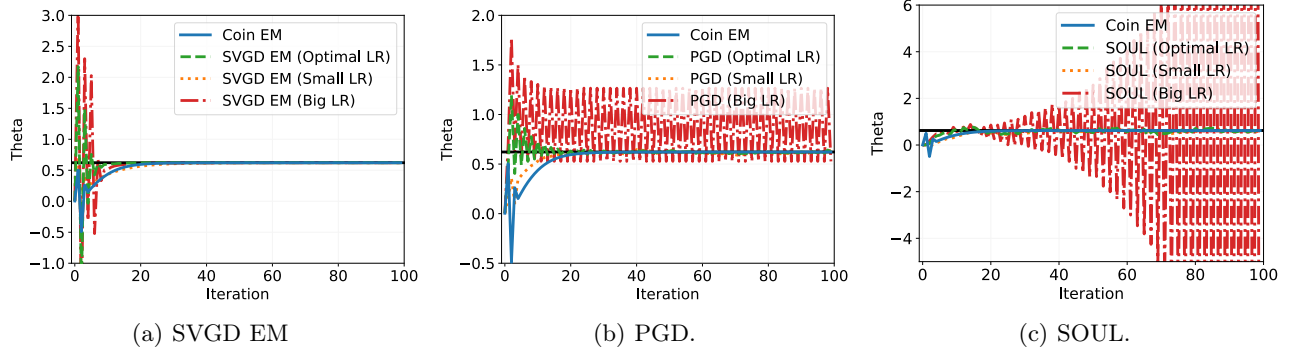


Figure 8: **Additional results for the toy hierarchical model with different learning rates.** The sequence of parameter estimates generated by SVGD EM, PGD, and SOUL, for three different learning rates: the ‘optimal’ learning rate from Fig. 7(a), a smaller learning rate, and a larger learning rate at the edge of stability.

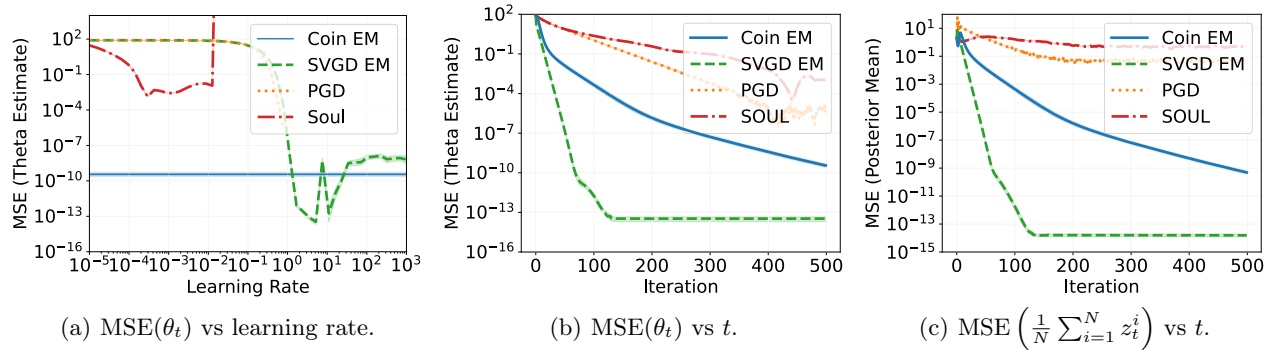


Figure 9: **Additional results for the toy hierarchical model with initialization $\theta_0 \sim \mathcal{N}(10, 0.1)$.** MSE of the parameter estimate θ_t as a function of the learning rate after $T = 500$ iterations (a); and MSE of the parameter estimate (b) and the posterior mean (c) as a function of the number of iterations, using the optimal learning rate from (a).

F.2 Bayesian logistic regression

In this section, we present additional numerical results for the Bayesian logistic regression model described in Sec. 4.2.

Additional results for different initializations. In Fig. 10, we plot the sequence of parameter estimates output by Coin EM, SVGD EM, PGD, PMGD, and SOUL, for different parameter initializations. In particular, we now initialize the parameter at $\theta_0 = 10$ or $\theta_0 = -10$, compared to $\theta_0 = 0$ in Fig. 3(a).

In both cases, all of methods converge to a similar value. SOUL is known to obtain accurate estimates of θ_* in this example De Bortoli et al. (2021), provided the learning rate is suitably small, and thus we use this as a benchmark. In these examples, where the parameter estimate is initialized far from the true parameter, the shorter transient exhibited by Coin EM relative to the other methods is even more evident.

Additional results for different learning rates. In Fig. 11, we plot the sequence of parameter estimates output by Coin EM, SVGD EM, PGD, PMGD, and SOUL, for different learning rates. In particular, we now use smaller learning rates of $\gamma = 0.001$ (PGD, PMGD, SOUL) or $\gamma = 0.005$ (SVGD EM), or larger learning rates of $\gamma = 0.1$ (PGD, PMGD, SOUL) or $\gamma = 0.5$ (SVGD EM). For reference, we also include the results for the original learning rates of $\gamma = 0.02$ (PGD, PMGD, SOUL) or $\gamma = 0.2$ (SVGD EM) in Fig. 3(a) (see Sec. E.2).

These figures illustrate the difficulties associated with tuning the learning rate for PGD, PMGD, SOUL and, to a lesser extent, SVGD EM. In particular, if the learning rate is chosen too small (Fig. 11(b)), then convergence

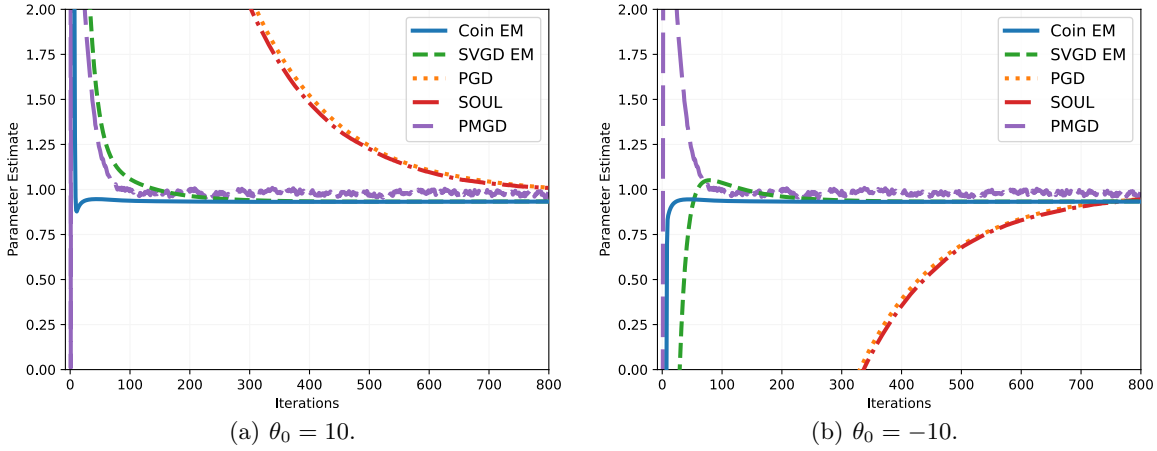


Figure 10: **Additional results for the Bayesian logistic regression in Sec. 4.2.** Parameter estimates θ_t from Coin EM, SVGD EM, PGD, SOUL, and PMGD, initialized at (a) $\theta_0 = 10$ and (b) $\theta_0 = -10$.

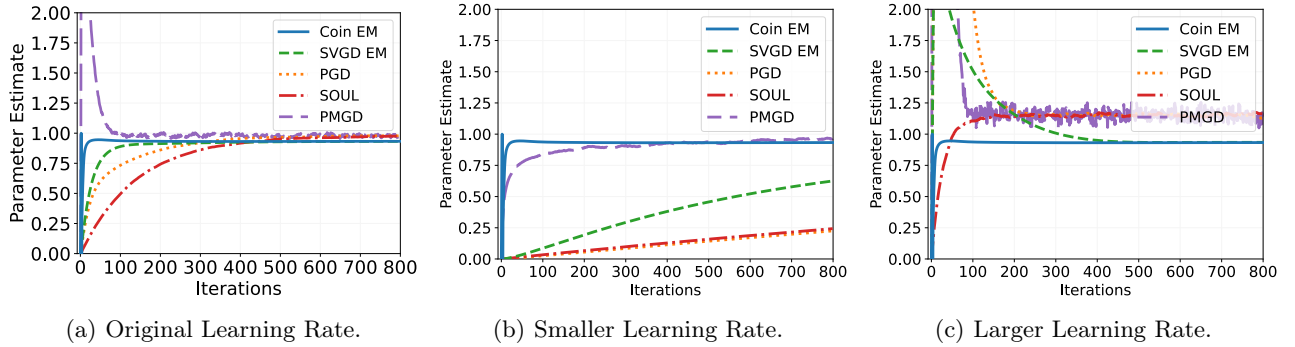


Figure 11: **Additional results for the Bayesian logistic regression in Sec. 4.2.** Parameter estimates θ_t from Coin EM, SVGD EM, PGD, SOUL, and PMGD, using (a) smaller learning rates $\gamma = 0.001$ (PGD, SOUL, PMGD) and $\gamma = 0.005$ (SVGD EM); and (b) larger learning rates $\gamma = 0.1$ (PGD, SOUL, PMGD) and $\gamma = 0.5$ (SVGD EM).

to the true parameter θ_* is painfully slow. On the other hand, if one selects a larger learning rate (Fig. 11(c)), convergence is more rapid, but one incurs a (significant) bias in the resulting parameter estimates. This is evident if one compares the asymptotic parameter estimates obtained in Fig. 11(a) and in Fig. 11(c). For PGD, PMGD, and SOUL, this bias originates in the bias associated with an Euler-Maruyama discretization of the Langevin dynamics; see, e.g., the discussion in (Kuntz et al., 2023, Sec. 2). Interestingly, SVGD EM does not seem to incur this bias to the same extent, despite the error associated with the finite-particle SVGD dynamics. Coin EM, of course, has no dependence on the learning rate, and is consistent across these experiments.

Additional results for different numbers of particles. Finally, we consider the impact of changing the number of particles used in the latent variable updates on the predictive performance. In Fig. 12, we repeat the experiment used to generate Fig. 3(c), but now using $N = 5, 20, 100$ particles. In this example, we see that there is little to be gained from increasing the number of particles in terms of the predictive performance, other than a minor increase in the performance of the learning-rate dependent methods (SVGD EM, PGD, SOUL) for sub-optimal choices of the learning rate. In this example, the posteriors are peaked and unimodal - see (De Bortoli et al., 2021, Fig. 2) - and can be approximated well using even a single particle in the vicinity of the modes.

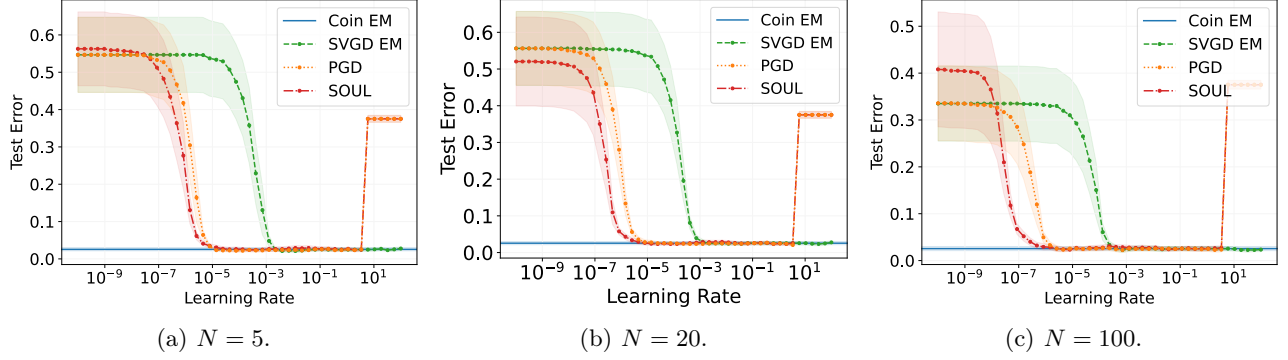


Figure 12: **Additional results for the Bayesian logistic regression in Sec. 4.2.** Test error as a function of the learning rate, for different numbers of particles.

F.3 Bayesian logistic regression (alternative model)

We now present numerical results for the alternative Bayesian logistic regression model described in App. E.3. In Fig. 13, we compare SVGD EM and Coin EM against PGD and SOUL, plotting the area under the receiver operator characteristic curve (AUC) as a function of the learning rate, after running each algorithm with 10 particles for 1000 iterations. For each of the datasets considered, the predictive performance of Coin EM is similar to the performance of SVGD EM, PGD and SOUL with well tuned learning rates. In comparison to the Bayesian logistic regression studied in Section 4.2, here the predictive performance of the SVGD EM, PGD, and SOUL is rather more sensitive to the learning rate, particularly for the Covtype dataset. In particular, in this case there is a much smaller range of values for which these methods exhibit performance on-par with, or superior to, Coin EM.

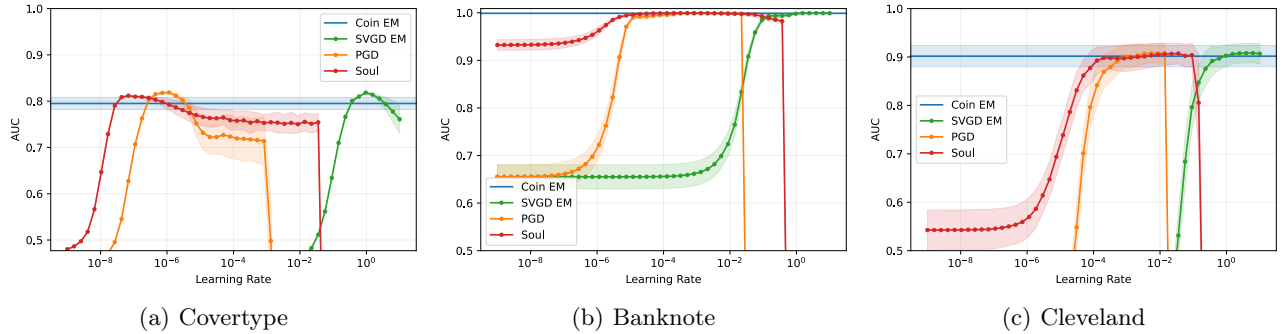


Figure 13: **Results for the alternative Bayesian logistic regression in App. E.3.** AUC as a function of the learning rate, averaged over 10 random test-train splits.

F.4 Bayesian neural network

We now present numerical results for the alternative Bayesian neural network model described in App. E.4.

In Fig. 14, we plot the test error achieved by SVGD EM, PGD, and SOUL for several different choices of learning rate; and for several different choices of the number of particles. For comparison, we also plot the test error achieved by Coin EM. We also include results for SVGD EM, PGD, and SOUL when using a scaling heuristic recommended in (Kuntz et al., 2023, Sec. 2), which is designed to stabilize the updates and avoid ill-conditioning. We refer to these methods as SVGD EM', PGD', and SOUL'.

Unsurprisingly, all methods other than Coin EM are highly dependent on the choice of learning rate. While, in all cases, we observe that the convergence rate can be improved by increasing the learning rate, this approach can only go so far. In particular, if the learning rate is increased much beyond the largest values considered in Figure 14, then the updates are likely to become unstable (see, e.g., Fig. 14(m) - 14(o)).

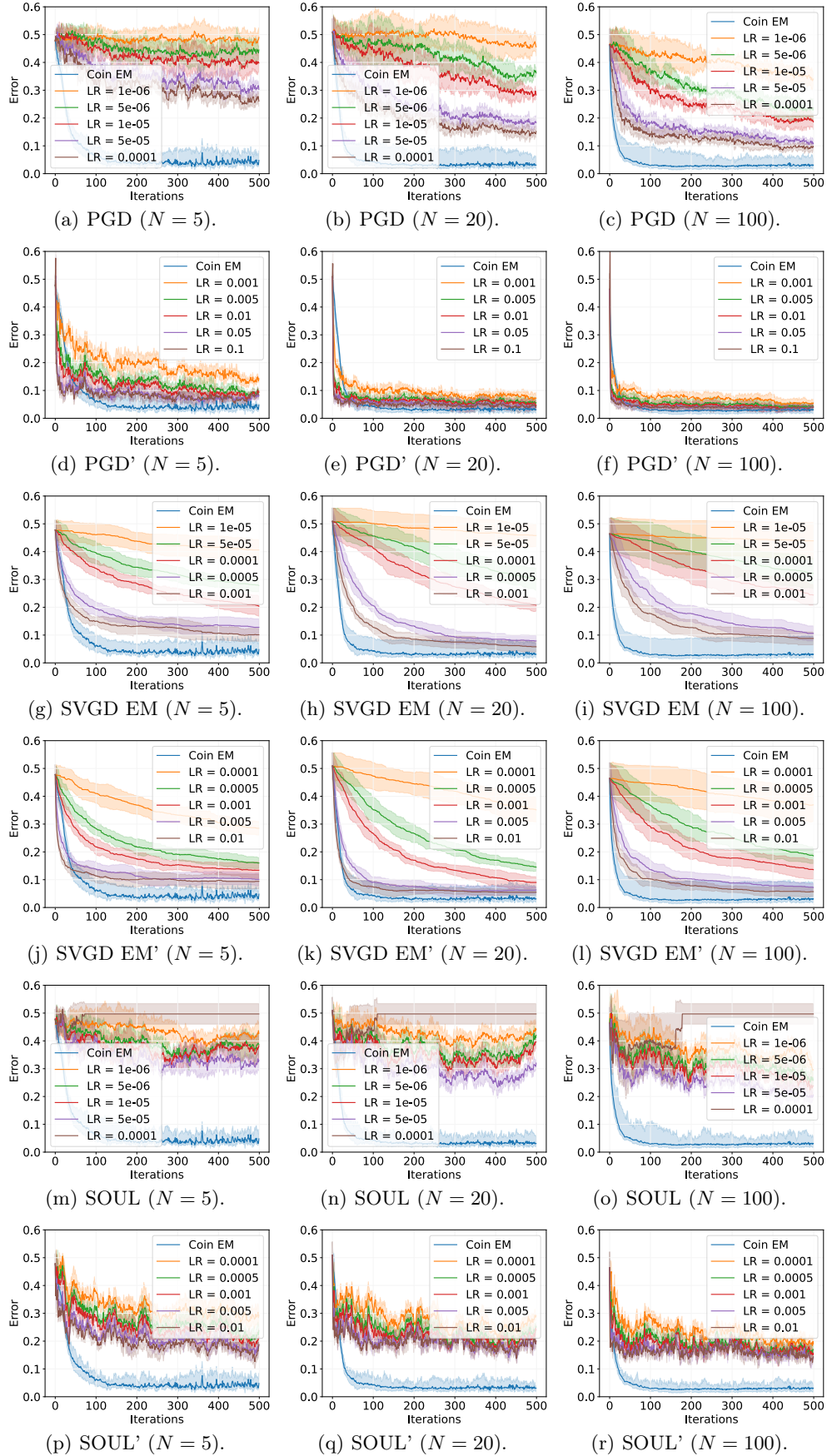


Figure 14: **Additional results for the Bayesian neural network model in Sec. 4.3: learning rate comparison.** Test error achieved over $T = 500$ training iterations, for different learning rates.

In Fig. 15, we further investigate the performance of each algorithm as we vary the number of particles. In this case, each method is improved by increasing the number of particles. Interestingly, Coin EM is robust to the number of particles, achieving good predictive performance even when the number is small (Fig. 15(a)). The same is true, to a lesser extent, for SVGD EM (with or without the heuristic), which performs relatively well across experiments (Fig. 15(b), Fig. 15(c)). PGD observes a more significant performance increase as the number of particles is increased (Fig. 15(d)), consistent with the observations in (Kuntz et al., 2023, Sec. 3.2), although performs relatively well even for small particular numbers with the heuristic (Fig. 15(e)). Finally, even for large numbers of particles, and when using the heuristic, SOUL struggles to compete with the other methods (Fig. 15(f), Fig. 15(g)).

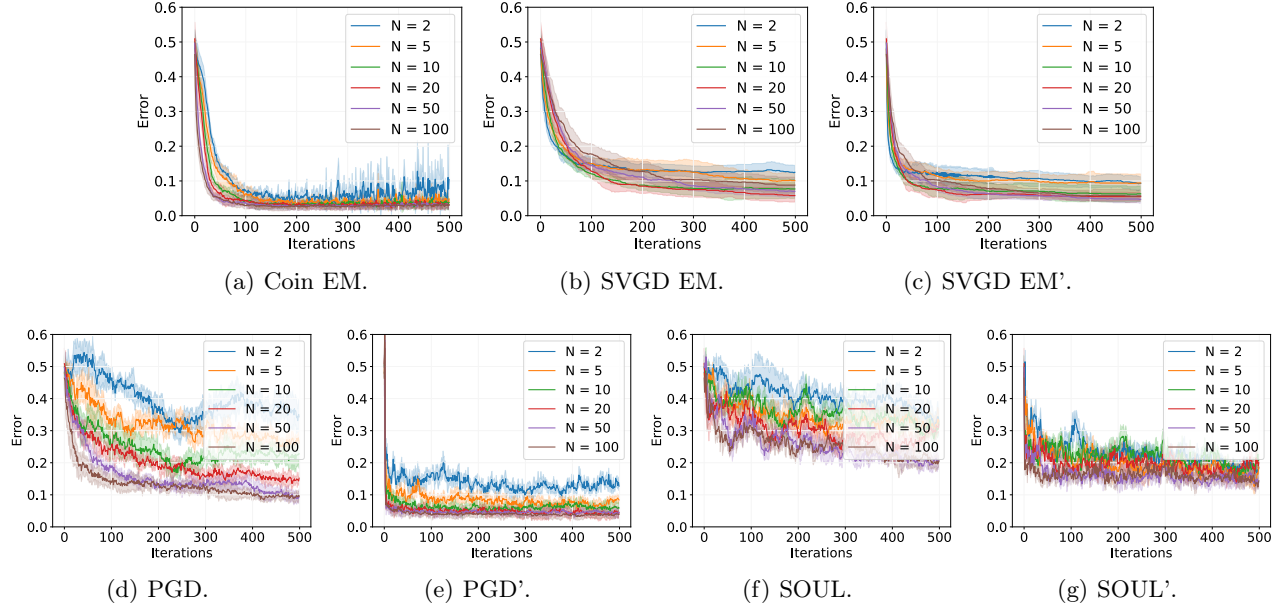


Figure 15: **Additional results for the Bayesian neural network model in Sec. 4.3: particle number comparison.** Test error over $T = 500$ training iterations, for different numbers of particles. For all learning-rate dependent methods, we use the best learning rate as determined by Fig. 14.

Finally, in Fig. 16, we provide a comparison of the test accuracy achieved by Coin EM, SVGD EM, PGD, and SOUL, now over a much finer grid of learning rates.

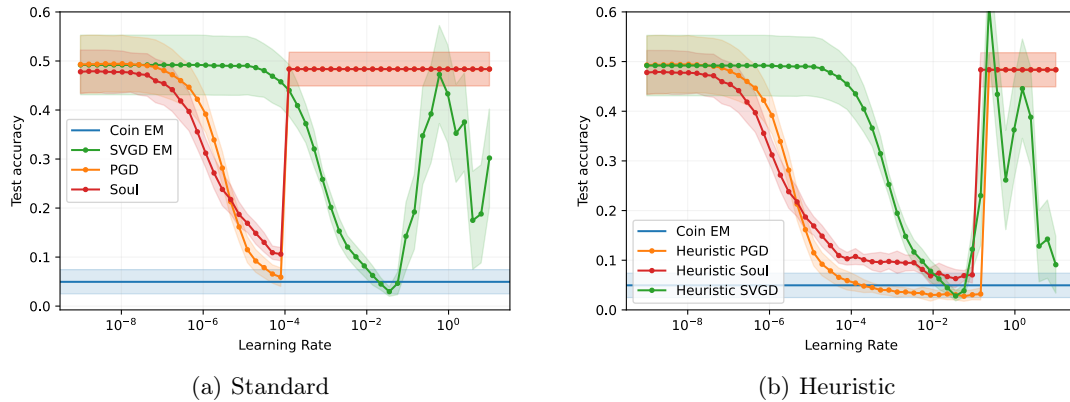


Figure 16: **Additional results for the Bayesian neural network in Sec. 4.3.** Accuracy as a function of the learning rate, for the MNIST dataset, averaged over ten random test-train splits.

F.5 Bayesian neural network (alternative model)

We now present numerical results for the alternative Bayesian neural network model described in App. E.5. In Fig. 17, we compare the performance of Coin EM and SVGD EM on several UCI benchmarks. Once again, we compare our algorithms with PGD and SOUL. As noted in Sec. 4.3, in this case the performance of the three learning-rate-dependent algorithms (SVGD EM, PGD, SOUL) is highly sensitive to the choice of learning rate. In particular, there is generally a very small range of learning rates for which these algorithms outperform Coin EM. Given an optimal choice of learning rate, the best predictive performance of SVGD EM is generally comparable with the best predictive performance of SOUL, and better than the best predictive performance of PGD, though there are cases in which Coin EM and SVGD EM significantly outperform the competing methods (see Fig. 17(f)).

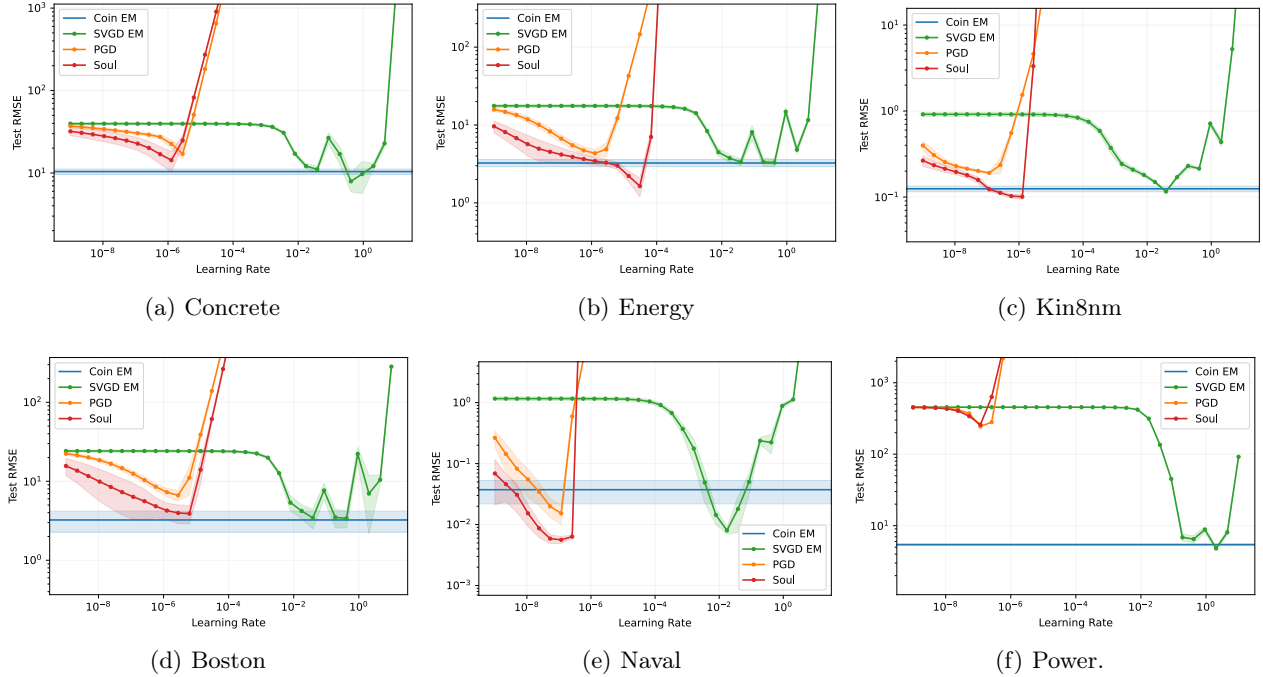


Figure 17: **Results for the alternative Bayesian neural network in App. E.5.** Root mean-squared-error (RMSE) as a function of the learning rate, for several UCI datasets, averaged over ten random test-train splits.

F.6 Network model

In this section, we provide additional results for latent space network model studied in Sec. 4.4. In Fig. 18 - Fig. 20, we plot the mean of the particles $\{z_T^i\}_{i=1}^N$ output by Coin EM (Fig. 18), PGD (Fig. 19), and SOUL (Fig. 20), using $N = 10$ particles and $T = 500$ iterations. In this case, each latent variable represents a network node, which corresponds to a Game of Thrones character. In each plot, Fig. (a) - (d) correspond to Series 1 - 4 of the TV series, respectively. The nodes are colour coded according to their cluster assignment from running DBScan Ester et al. (1996).

In this case, only Coin EM is able to successfully capture the subjectively correct relationships between characters. We experimented with various learning rates for PGD and SOUL and were not able to improve the latent representation of the nodes to infer useful clusters among the characters. If additional covariate information were available, e.g. house labels such as Targaryen, Lannister, etc. then it is possible that the additional information could improve the latent representation for the PGD and SOUL algorithms.

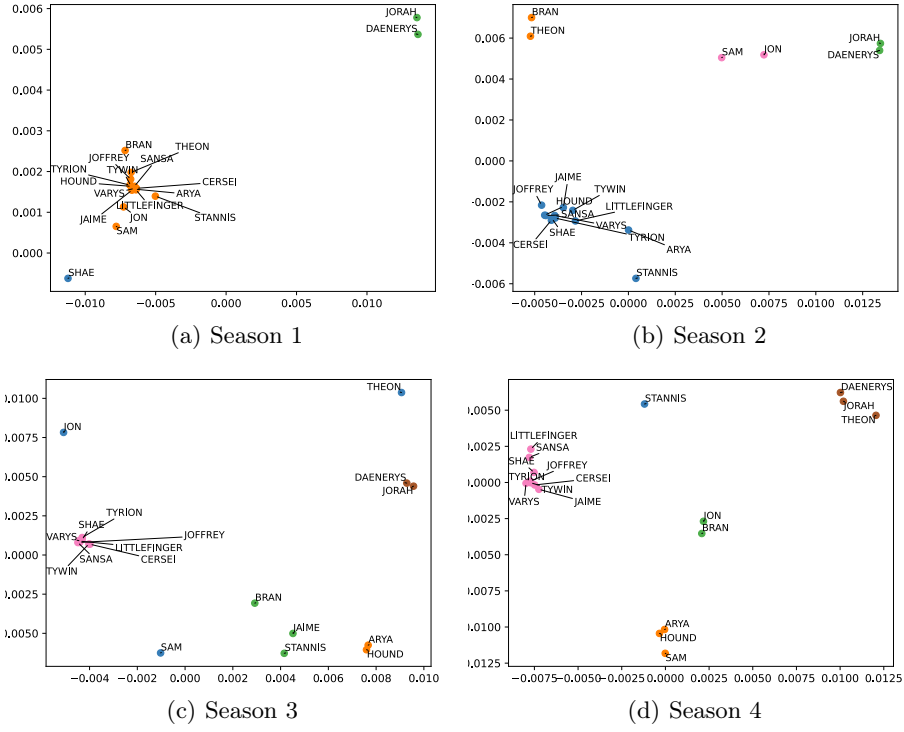


Figure 18: **Additional results for the network model in Sec. 4.4.** Posterior mean $\frac{1}{N} \sum_{j=1}^N z_T^j$ of the particles generated by Coin EM after $T = 500$ iterations.

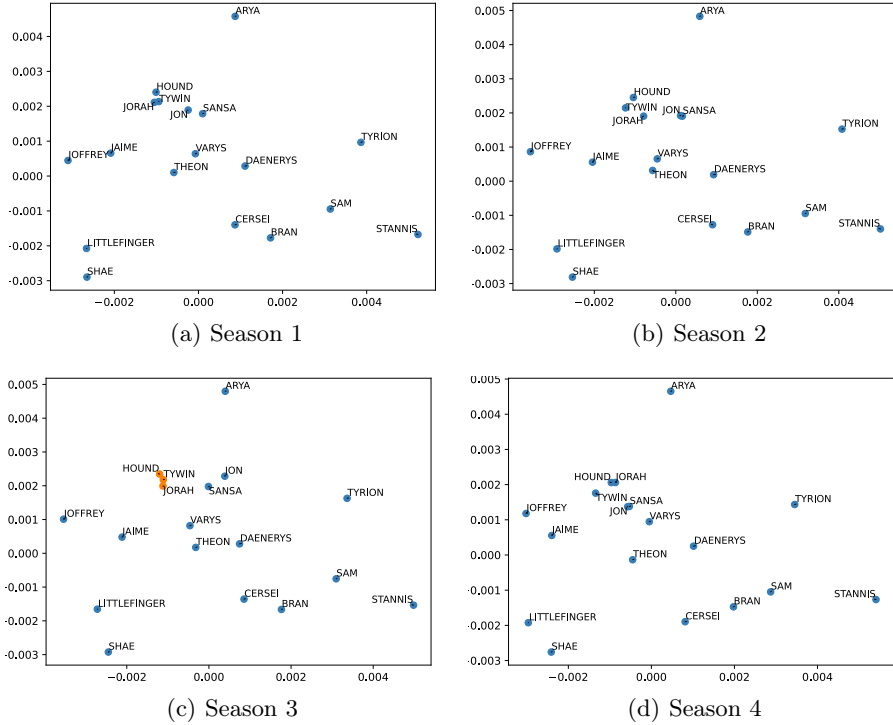


Figure 19: **Additional results for the network model in Sec. 4.4.** Posterior mean $\frac{1}{N} \sum_{j=1}^N z_T^j$ of the particles generated by PGD after $T = 500$ iterations.

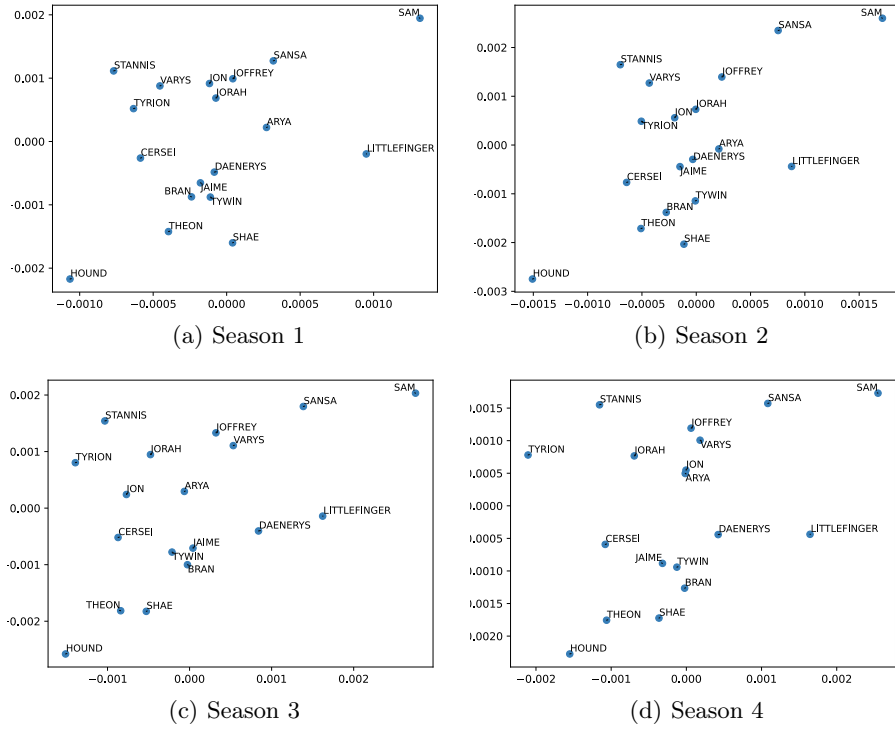


Figure 20: **Additional results for the network model in Sec. 4.4.** Posterior mean $\frac{1}{N} \sum_{j=1}^N z_T^j$ of the particles generated by SOUL after $T = 500$ iterations.