Sampling from Bayesian Neural Network Posteriors with Symmetric Minibatch Splitting Langevin Dynamics

Daniel Paulin* Nanyang Technological U.
Peter A. Whalley* ETH Zürich
Neil K. Chada City University of Hong Kong
Benedict Leimkuhler University of Edinburgh

Abstract

We propose a scalable kinetic Langevin dynamics algorithm for sampling parameter spaces of big data and AI applications. Our scheme combines a symmetric forward/backward sweep over minibatches with a symmetric discretization of Langevin dynamics. For a particular Langevin splitting method (UBU), we show that the resulting Symmetric Minibatch Splitting-UBU (SMS-UBU) integrator has bias $\mathcal{O}(h^2 d^{1/2})$ in dimension $d > 0$ with stepsize $h > 0$, despite only using one minibatch per iteration, thus providing excellent control of the sampling bias as a function of the stepsize. We apply the algorithm to explore local modes of the posterior distribution of Bayesian neural networks (BNNs) and evaluate the calibration performance of the posterior predictive probabilities for neural networks with convolutional neural network architectures for classification problems on three different datasets (Fashion-MNIST, Celeb-A and chest X-ray). Our results indicate that BNNs sampled with SMS-UBU can offer significantly better calibration performance compared to standard methods of training and stochastic weight averaging.

1 INTRODUCTION

Bayesian neural networks (BNNs) are a statistical paradigm which has been widely advocated to improve the reliability of neural networks by formulating questions regarding parameter sensitivity and prediction error in terms of the structure of the posterior parameter distribution. The ultimate goals of BNNs (already espoused in the seminal works of Neal (2012) and MacKay (1995)) are to reduce generalization error and enhance robustness of trained models. Due to the capacity of BNNs to quantify uncertainty, they have been proposed for use in applications such as astronomy, and modelling partial differential equations (Cranmer et al. (2021), Geneva and Zabaras (2020)), but their potential impact is much greater and includes topics in data-driven engineering (Levinson and et al. (2011), Thadeshwar et al. (2020)) and automated medical diagnosis (Band et al. (2021), Koh et al. (2022)).

While reliance on a Bayesian statistics approach provides theoretical foundation for BNNs, the high cost of treating these models has hampered uptake in practical applications. In the modern era, datasets are often very large–a single epoch (pass through the training data) with computation of likelihoods for a large scale network may take hours or days (Shen et al. (2023)). Optimization schemes can robustly use efficient data subsampling, whereas BNNs require a more delicate approach, since the target distribution of the Markov chain is easily corrupted (Chen et al. (2014), Cobb and Jalaian (2021), Zhang et al. (2020)). Scalable alternatives to MCMC based on Gaussian approximation can lead to some improvements in calibration (Maddox et al. (2019), Blundell et al. (2015)), but they lack theoretical justification.

The high computational cost of deep learning is also a consequence of the size of the model parameter space, which may consist of many millions (or even billions) of parameters. Sampling the parameters of such models undeniably adds substantial overload in comparison to optimization, which is already seen as costly, thus a central impediment to exploiting the promise of BNNs is the need for optimal sampling strategies that scale well both in terms of the size of the dataset and parameter space dimension (Dusenberry et al. (2020), Khan et al. (2021)). The most popular schemes for high dimensional inference are Markov chain Monte Carlo (MCMC) methods (Andrieu et al. (2003)), e.g. Hamiltonian Monte Carlo (HMC) and the Metropolis-adjusted Langevin algorithm (MALA) (Bardenet et al. (2017), Chen et al. (2014), Gouraud et al. (2025)), with HMC typically favored in the large scale BNN setting (Izmailov et al. (2021), Papamarkou et al. (2022)).

In this article, we present the use of an "unadjusted" sampling scheme, within BNNs, which avoids the costly Metropolis test and is based on the use of efficient sym-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

metric splitting methods for kinetic Langevin dynamics, in particular the UBU discretization, providing second order accuracy in the strong (pathwise) sense and in terms of sampling bias. We show that second order accuracy can be maintained in combination with a carefully organized minibatch subsampling strategy resulting in a fully symmetric (SMS-UBU) integrator, which provides both efficiency for large datasets and low bias, asymptotically, at high dimension. We implement this scheme for BNNs and demonstrate the performance in comparison with some alternative methods. Numerical tests are conducted using suitable NNs on a range of classification problems involving image data (the fashion-MNIST dataset (Xiao et al. (2017)), celeb-A dataset (Yang et al. (2015)) and a chest X-ray dataset (Kermany et al. (2018))). To assess the results, we rely on well-known metrics, namely accuracy, negative log-likelihood on test data, adaptive calibration error, and ranked probability score. Our experiments clearly demonstrate the effectiveness of BNNs trained with SMS-UBU compared to standard training and stochastic weight averaging. Our code is available at `github.com/paulindani/SMS_Kinetic_Langevin`.

The main contributions of this work are: (i) the presentation of a kinetic Langevin scheme incorporating symmetric minibatch subsampling (SMS-UBU) and the demonstration of its second order accuracy, the first result of its kind for a stochastic gradient scheme, (ii) novel bias bounds for vanilla stochastic gradient kinetic Langevin dynamics as a function of dataset size, considering also the control of the bias for minibatches of the dataset drawn without replacement, and (iii) demonstration of the use of a Bayesian uncertainty quantification (UQ) framework for NNs by applying our efficient methods to sample BNNs near targeted posterior modes. In Bayesian uncertainty quantification, we estimate the uncertainties in the predictions by sampling from the Bayesian posterior distribution of the model's parameters.

The results are organized as follows: Section 2 gives an overview of kinetic Langevin integrators, while the SMS-UBU schemes appears in Section 2.3 along with a summary of the key theoretical results. Our main theorem is given in Section 3. Section 4 takes up discussion of UQ in neural networks, followed by the presentation of classification experiments in Section 5. We conclude in Section 6. Detailed proofs are provided in the supplement, along with additional details about the experiments.

2 Methodology

In this section, we first review sampling using kinetic Langevin integrators. We begin with a brief overview, before discussing a particular splitting scheme, (UBU). We will then discuss the extension of UBU to stochastic gradients before introducing a new method referred to as SMS-UBU, which is presented in algorithmic form.

2.1 Kinetic Langevin dynamics and discretizations

Denote by $\pi(dx)$ the target probability measure of form $\pi(dx) \propto \exp(-f(x))dx$, where $f(x)$ refers to the potential energy function. Kinetic Langevin dynamics is the system

$$\begin{aligned} dX_t &= V_t dt, \\ dV_t &= -\nabla f(X_t)dt - \gamma V_t dt + \sqrt{2\gamma} dW_t, \end{aligned} \tag{1}$$

on $\mathbb{R}^{2d}$, where $\{W_t\}_{t\geq 0}$ is a standard $d-$dimensional Wiener process, and $\gamma > 0$ is a friction coefficient. Under standard assumptions (see Pavliotis (2014)), the unique invariant measure of the process $\{X_t, V_t\}_{t\geq 0}$ has density given by

$$\begin{aligned} \bar{\pi}(dx, dv) &\propto \exp\left(-f(x) - \frac{\|v\|^2}{2}\right) dxdv \\ &= \pi(dx) \exp\left(-\frac{\|v\|^2}{2}\right) dv, \end{aligned} \tag{2}$$

with respect to Lebesgue measure. Due to the product form of the invariant measure, averages with respect to the target measure $\pi(dx)$ can be obtained as the configurations generated by approximating paths of (1).

Bias in the invariant measure arises, principally, due to two factors: (i) finite stepsize discretization of (1), and (ii) the use of stochastic gradients. The choice of integrator, the stepsize, and the friction γ all affect both the convergence rate and the discretization bias (Bou-Rabee and Owhadi (2010), Leimkuhler and Matthews (2013), Gouraud et al. (2025)), and this is further altered by the procedure used to estimate the gradient. While the simplest procedure is to use the Euler-Mayurama scheme for the overdamped form of (1), together with stochastic gradients (SG), as in the SG-HMC algorithm of Chen et al. (2014), there has been significant progress in the numerical analysis community in the development of accurate second order integrators for kinetic Langevin dynamics (1), see Leimkuhler and Matthews (2013), Sanz-Serna and Zygalakis (2021) and Alfonso Álamo (2021). Here we extend some of these integrators via the use of stochastic gradients.

An accurate splitting method was introduced in Alfonso Álamo (2021) and further studied in Sanz-Serna and Zygalakis (2021) and Chada et al. (2023). This splitting method only requires one gradient evaluation per iteration but has strong order two. The method is based on breaking up the SDE (1) as follows

$$\begin{pmatrix} dX_t \\ dV_t \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ -\nabla f(X_t)dt \end{pmatrix}}_{\mathcal{B}} + \underbrace{\begin{pmatrix} V_t dt \\ -\gamma V_t dt + \sqrt{2\gamma} dW_t \end{pmatrix}}_{\mathcal{U}},$$

where each part can be integrated exactly over a step of size h. Given $\gamma > 0$, let $\eta = \exp(-\gamma h/2)$, and for ease of notation, define the following operators

$$\mathcal{B}(x, v, h) = (x, v - h\nabla f(x)), \tag{3}$$

and

$$\mathcal{U}(x,v,h/2,\xi^{(1)},\xi^{(2)}) = \Big(x + \frac{1-\eta}{\gamma}v \tag{4}$$
$$+ \sqrt{\frac{2}{\gamma}}\left(\mathcal{Z}^{(1)}\left(h/2,\xi^{(1)}\right) - \mathcal{Z}^{(2)}\left(h/2,\xi^{(1)},\xi^{(2)}\right)\right),$$
$$\eta v + \sqrt{2\gamma}\mathcal{Z}^{(2)}\left(h/2,\xi^{(1)},\xi^{(2)}\right)\Big), \text{ where}$$
$$\mathcal{Z}^{(1)}\left(h/2,\xi^{(1)}\right) = \sqrt{\frac{h}{2}}\xi^{(1)}, \tag{5}$$
$$\mathcal{Z}^{(2)}\left(h/2,\xi^{(1)},\xi^{(2)}\right) = \tag{6}$$
$$\sqrt{\frac{1-\eta^2}{2\gamma}}\left(\sqrt{\frac{1-\eta}{1+\eta}\cdot\frac{4}{\gamma h}}\xi^{(1)} + \sqrt{1-\frac{1-\eta}{1+\eta}\cdot\frac{4}{\gamma h}}\xi^{(2)}\right),$$

where $\xi^{(1)},\xi^{(2)} \sim \mathcal{N}(0_d, I_d)$ are d-dimensional standard Gaussian random variables.

The UBU scheme consists of applying first a half-step (h replaced by $h/2$) using (4), then applying an impulse based on the potential energy term, and following this by another half-step of the mapping $\mathcal{U}$. The symmetry of this scheme plays a role in its accuracy, since it induces a cancellation of the leading order term in the error expansion (in the absence of gradient noise).

Other symmetric splitting methods are possible and have been extensively studied in recent years. In particular, the BAOAB, ABOBA and OBABO schemes (Bussi and Parrinello (2007), Leimkuhler and Matthews (2013), Leimkuhler et al. (2016)) are all second order in the weak (sampling bias) sense and make plausible candidates for BNN sampling. These methods can be viewed as breaking the $\mathcal{U}$ stage of the UBU algorithm into two parts and interspersing the resulting solution maps with $\mathcal{B}$ steps to form a numerical scheme. Several of these methods will be compared in Section 5.

2.2 Stochastic gradient algorithms

We first abstractly define stochastic gradients as unbiased estimators of the gradient of the potential, under the same set of assumptions as Leimkuhler et al. (2024a).

Definition 1. *A stochastic gradient approximation of a potential f is defined by a function $\mathcal{G} : \mathbb{R}^d \times \Omega \to \mathbb{R}^d$ and a probability distribution ρ on a Polish space Ω, such that for every $x \in \mathbb{R}^d$, $\mathcal{G}(x,\cdot)$ is measurable on $(\Omega,\mathcal{F})$, and for $\omega \sim \rho$,*

$$\mathbb{E}(\mathcal{G}(x,\omega)) = \nabla f(x).$$

The function $\mathcal{G}$ and the distribution ρ together define the stochastic gradient, which we denote as $(\mathcal{G},\rho)$.

The following assumption is useful for controlling the accuracy of the stochastic gradient approximations.

Assumption 1. The Jacobian of the stochastic gradient $\mathcal{G}$, $D_x\mathcal{G}(x,\omega)$ exists and is measurable on $(\Omega,\mathcal{F})$. There exists a bound $C_G > 0$ such that, for $\omega \sim \rho$,

$$\sup_{x\in\mathbb{R}^d} \mathbb{E}\|D_x\mathcal{G}(x,\omega) - \nabla^2 f(x)\|^2 \le C_G.$$

Assumption 2 (Moments of Stochastic Gradient). Let $\mathcal{D}(y,\omega) := \mathcal{G}(y,\omega) - \nabla f(y)$, for all $y \in \mathbb{R}^d$ be the difference between the stochastic gradient approximation, $\mathcal{G}(y,\omega)$, and the true gradient. The following moment bound holds on $\mathcal{D}$

$$\mathbb{E}\left[\|\mathcal{D}(y,\omega)\|^2 \mid y\right] \le C_{SG}^2 M^2 \|y - x^*\|^2.$$

In Bayesian inference settings, stochastic gradients are usually considered for a potential $f : \mathbb{R}^d \to \mathbb{R}$ of the form

$$f(x) = f_0(x) + \sum_{i=1}^{N_D} f_i(x), \tag{7}$$

where $x \in \mathbb{R}^d$ and N_D is the size of the dataset; the individual contributions arise as terms in a summation that defines the log-likelihood of parameters given the data. In many applications, such as the ones we consider in this work, both the dimension d and the size of the dataset N_D are large. As a result, gradient evaluations of the potential can be computationally expensive. As a consequence, a stochastic approximation of the gradient is typically used (see e.g. Robbins and Monro (1951)) which dramatically reduces the computational cost by relying only on a sample of the dataset of size $N_b \ll N_D$. For MCMC sampling it is natural to seek a similar approach to improve the scalability (Welling and Teh (2011), Nemeth and Fearnhead (2021)). Within (7), f_0 can be chosen to be the negative log density of the prior distribution. Then random variables of the form $\omega \in [N_D]^{N_b}$ are constructed as uniform draws on $[N_D] = \{1,\dots,N_D\}$, taken i.i.d. with replacement (Baker et al. (2019)). At each step one uses an unbiased estimator of (7) instead of the full gradient evaluation, for example

$$\mathcal{G}(x,\omega|\hat{x}) = \nabla f_0(x) + \sum_{i=1}^{N_D}\nabla f_i(\hat{x}) + \frac{N_D}{N_b}\sum_{i\in\omega}[\nabla f_i(x) - \nabla f_i(\hat{x})], \tag{8}$$

such that $\mathbb{E}(\mathcal{G}(x,\omega)) = \nabla f(x)$, where, in case the potential is convex, $\hat{x}$ can be chosen as the minimizer of f, $x^* \in \mathbb{R}^d$. We define the random variables

$$Z^x := \sqrt{\frac{2}{\gamma}}\left(\mathcal{Z}^{(1)}\left(h/2,\xi^{(1)}\right) - \mathcal{Z}^{(2)}\left(h/2,\xi^{(1)},\xi^{(2)}\right)\right),$$
$$Z^v := \sqrt{2\gamma}\mathcal{Z}^{(2)}\left(h/2,\xi^{(1)},\xi^{(2)}\right), \tag{9}$$

based on (5) and (6). In Algorithm 1 we formulate a stochastic gradient implementation of the UBU scheme.

Algorithm 1 Stochastic Gradient UBU (SG-UBU)

Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$, sample size $K > 0$ and friction parameter $\gamma > 0$.
for $k = 1, 2, ..., K$ **do**
  Sample $Z_k^x, Z_k^v, \tilde{Z}_k^x, \tilde{Z}_k^v$ according to (9)
(U) $(x, v) \to (x_{k-1} + \frac{1-\eta^{1/2}}{\gamma} v_{k-1} + Z_k^x, \eta^{1/2} v_{k-1} + Z_k^v)$
  Sample $\omega_k \sim \rho$
(B) $v \to v - h\mathcal{G}(x, \omega_k)$
(U) $(x_k, v_k) \to (x + \frac{1-\eta^{1/2}}{\gamma} v + \tilde{Z}_k^x, \eta^{1/2} v + \tilde{Z}_k^v)$
end for
Output: Samples $(x_k)_{k=0}^K$.

A limitation of using stochastic gradients is that it can introduce additional bias. In particular, using the standard stochastic gradient approximation (sampling the minibatches i.i.d. at each iteration) reduces the strong order of a numerical integrator for kinetic Langevin dynamics to $O(h^{1/2})$. SG-UBU for example will be order $1/2$, a dramatic reduction in accuracy compared to its full gradient counterpart, which has strong order two (see Alfonso Álamo (2021), Sanz-Serna and Zygalakis (2021)). In terms of the weak order and bias in empirical averages stochastic gradients reduce the order of accuracy in the stepsize from two to one (Vollmer et al. (2016), Sekkat and Stoltz (2023)).

A consequence of the reduced accuracy with respect to stepsize is substantially worse non-asymptotic guarantees and scaling in terms of dimension (Chatterji et al. (2018), Dalalyan and Karagulyan (2019), Gouraud et al. (2025)). We note that some practical implementations in the literature of stochastic gradient based sampling schemes also consider sampling without replacement, as discussed in Detommaso et al. (2023). We do not prove theoretical results for such a version of SG-UBU, but perform some numerical experiments in Sections 3.4 and 5.

2.3 Symmetric Minibatch Splitting UBU (SMS-UBU)

We now introduce an alternative stochastic gradient approximation method that relies on a random partition $\omega_1, ..., \omega_{N_m}$ of $[N_D] = \{1, \dots, N_D\}$, each of size N_b (i.e. they are sampled uniformly without replacement from $[N_D]$, and $\omega_1 \cup \ldots \cup \omega_{N_m} = [N_D]$). We propose to take UBU steps with gradient approximations using the index-set ω_1 first, then ω_2, etc., all the way up to ω_{N_m}. We follow this with steps using the same sequence of minibatches taken in the reverse order. We illustrate this methodology for the UBU integrator below:

$$\underbrace{(\mathcal{U}\mathcal{B}_{\omega_1}\mathcal{U})}_{\text{minibatch 1}} \dots \underbrace{(\mathcal{U}\mathcal{B}_{\omega_{N_m}}\mathcal{U})}_{\text{minibatch } N_m} \underbrace{(\mathcal{U}\mathcal{B}_{\omega_{N_m}}\mathcal{U})}_{\text{minibatch } N_m} \dots \underbrace{(\mathcal{U}\mathcal{B}_{\omega_1}\mathcal{U})}_{\text{minibatch 1}},$$

where

$$\mathcal{B}_{\omega_l}(x, v, h) = \left(x, v - hN_m \sum_{i \in \omega_l} \nabla f_i(x) \right),$$

and $N_m := N_D/N_b$ is the number of minibatches. This framework is not specific to the UBU integrator and can be applied to any kinetic Langevin dynamics integrator. However, we detail precisely the SMS stochastic gradient algorithm for the UBU integrator in Algorithm 2.

A motivation for such a splitting is that it ensures that the integrator is symmetric as an integrator on a longer time horizon $T = 2hN_m$ and therefore has improved weak order properties Leimkuhler et al. (2016). It also turns out that it gains improved strong order properties, which we make rigorous in the following sections.

Remark 1. A similar minibatch selection procedure was used within the context of HMC to integrate Hamiltonian dynamics based on symmetric splitting with stochastic gradient approximations for each leapfrog step (Cobb and Jalaian (2021)). Their motivation was different than ours: as they were working in the HMC setting, they required an integrator for Hamiltonian dynamics that offered high Metropolis-Hastings acceptance rate. They did not study the use of this type of subsampling for unadjusted Langevin. Their approach can only take samples after a complete forward/backward sweep (going through the entire dataset) when the accept/reject step is applied. This requires going through the entire dataset between consecutive samples, which is not needed for unadjusted Langevin. We numerically evaluate such an HMC-based approach in Section F of the Supplementary material.

Remark 2. A similar procedure is proposed in Franzese et al. (2022), also considering back and forth operators. That paper studies a variety of numerical methods, including OABAO, a leapfrog scheme, and a higher order scheme, but as pointed out in their paper the order of accuracy is limited by the error due to gradient noise. Although they demonstrate an improved accuracy with the higher order scheme it requires multiple gradient evaluations per sample, which is considerably less efficient than our approach.

3 Theory

In this section we present theoretical results related to SMS-UBU. We provide the key assumptions to establish a Wasserstein contraction result related to the convergence of our numerical scheme, and we give a global error estimate which establishes the $\mathcal{O}(h^2)$ strong accuracy bound, where h is the stepsize used. Proofs of these results are deferred to the appendix.

3.1 Assumptions

We first state the assumptions used in our proofs.

Assumption 3. f is m-strongly convex if there exists a $m > 0$ such that for all $x, y \in \mathbb{R}^d$

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq m \left| x - y \right|^2.$$

Algorithm 2 Symmetric Minibatch Splitting UBU (SMS-UBU)

Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$, sample size $K > 0$, friction parameter $\gamma > 0$ and number of minibatches N_m.
for $i = 1, 2, ..., \lceil K/2N_m \rceil$ **do**
Sample $\omega_1, ..., \omega_{N_m} \in [N_D]^{N_b}$ uniformly without replacement.

Forward Sweep

for $k = 1, 2, ..., \min\{N_m, K - (2i-2)N_m\}$ **do**
Sample $Z_k^x, Z_k^v, \tilde{Z}_k^x, \tilde{Z}_k^v$ according to (9)
(U) $(x, v) \to (x_{(2i-2)N_m+k-1} + \frac{1-\eta^{1/2}}{\gamma} v_{(2i-2)N_m+k-1} + Z_k^x, \eta^{1/2} v_{(2i-2)N_m+k-1} + Z_k^v)$
(B) $v \to v - h\mathcal{G}(x, \omega_k)$
(U) $(x_{2N_m i+k}, v_{2N_m i+k}) \to (x + \frac{1-\eta^{1/2}}{\gamma} v + \tilde{Z}_k^x, \eta^{1/2} v + \tilde{Z}_k^v)$
end for

Backward Sweep

for $k = 1, 2, ..., \min\{N_m, K - (2i-1)N_m\}$ **do**
Sample $Z_k^x, Z_k^v, \tilde{Z}_k^x, \tilde{Z}_k^v$ according to (9)
(U) $(x, v) \to (x_{(2i-1)N_m+k-1} + \frac{1-\eta^{1/2}}{\gamma} v_{(2i-1)N_m+k-1} + Z_k^x, \eta^{1/2} v_{(2i-1)N_m+k-1} + Z_k^v)$
(B) $v \to v - h\mathcal{G}(x, \omega_{N_m+1-k})$
(U) $(x_{(2i-1)N_m+k}, v_{(2i-1)N_m+k}) \to (x + \frac{1-\eta^{1/2}}{\gamma} v + \tilde{Z}_k^x, \eta^{1/2} v + \tilde{Z}_k^v)$
end for
end for
Output: Samples $(x_k)_{k=0}^K$.

Assumption 4. $f : \mathbb{R}^d \to \mathbb{R}$ is C^2, and there exists $M > 0$ such that for all $x, y \in \mathbb{R}^d$

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|.$$

Assumption 5. The potential $f : \mathbb{R}^d \to \mathbb{R}$ is C^3 and there exists $M_1 > 0$ such that for all $x, y \in \mathbb{R}^d$,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq M_1\|x - y\|,$$

this implies that

$$\|\nabla^3 f(x)[v, v']\| \leq M_1\|v\|\|v'\|,$$

as used in Sanz-Serna and Zygalakis (2021).

The strongly Hessian Lipschitz property relies on a specific tensor norm from Chen and Gatmiry (2023), which we use to establish improved dimension dependence. A small strongly Hessian Lipschitz constant for $\nabla^3 f$ is shown to hold for some applications of practical interest (see Chen and Gatmiry (2023), Chada et al. (2023) and Lemma 7 of Paulin and Whalley (2024)).

Definition 2. *For $A \in \mathbb{R}^{d\times d\times d}$, let*

$$\|A\|_{\{12\}\{3\}} = \left\| \sum_{i_1} A_{i_1,\cdot,\cdot}^T \cdot A_{i_1,\cdot,\cdot} \right\|^{1/2}, \tag{10}$$

where $\|\cdot\|$ refers to the L^2 matrix norm, and $A_{i_1,\cdot,\cdot} = (A_{i_1,i_2,i_3})_{1\leq i_2\leq d, 1\leq i_3\leq d}$ is a $d \times d$ matrix.

Assumption 6 (M_1^s-strongly Hessian Lipschitz). $f : \mathbb{R}^d \to \mathbb{R}$ is three times continuously differentiable and M_1^s-strongly Hessian Lipschitz if there exists $M_1^s > 0$ such that

$$\|\nabla^3 f(x)\|_{\{1,2\}\{3\}} \leq M_1^s,$$

for all $x \in \mathbb{R}^d$.

3.2 Convergence of the numerical method

In this subsection, we summarize Wasserstein convergence results from Chada et al. (2023) which used the technique developed in Leimkuhler et al. (2024c). There have been many works on couplings of kinetic Langevin dynamics and its discretization including Cheng et al. (2018), Dalalyan and Riou-Durand (2020), Monmarché (2021), Sanz-Serna and Zygalakis (2021), Chak and Monmarché (2023) and Schuh and Whalley (2024) in the discretized setting and Eberle et al. (2019) and Schuh (2024) in the continuous setting. We remark that Wasserstein convergence for the UBU discretization was first studied in Sanz-Serna and Zygalakis (2021).

Definition 3 (Weighted Euclidean norm)**.** *For $z = (x, v) \in \mathbb{R}^{2d}$ the weighted Euclidean norm of z is defined by*

$$||z||_{a,b}^2 = ||x||^2 + 2b\langle x, v\rangle + a||v||^2,$$

for $a, b > 0$ with $b^2 < a$.

Remark 3. Using the assumption $b^2 < a$, we can show that this is equivalent to the Euclidean norm on $\mathbb{R}^{2d}$. Under the condition $b^2 \leq a/4$, we have

$$\frac{1}{2}\min(a, 1)\|z\|^2 \leq \frac{1}{2}||z||_{a,0}^2 \leq ||z||_{a,b}^2 \leq \frac{3}{2}||z||_{a,0}^2 \leq \frac{3}{2}\max(a, 1)\|z\|^2. \tag{11}$$

Definition 4 (p-Wasserstein distance)**.** *Let us define $\mathcal{P}_p(\mathbb{R}^{2d})$ to be the set of probability measures which have p-th moment for $p \in [1, \infty)$ (i.e. $\mathbb{E}(\|Z\|^p) < \infty$). Then the p-Wasserstein distance in norm $\|\cdot\|_{a,b}$ between two measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^{2d})$ is defined as*

$$\mathcal{W}_{p,a,b}(\nu, \mu) = \left(\inf_{\xi\in\Gamma(\nu,\mu)} \int_{\mathbb{R}^{2d}} \|z_1 - z_2\|_{a,b}^p d\xi(z_1, z_2) \right)^{1/p}, \tag{12}$$

where $\|\cdot\|_{a,b}$ is the norm introduced before and that $\Gamma(\nu, \mu)$ is the set of measures with respective marginals of ν and μ.

Remark 4. We use $\mathcal{W}_2$ to mean the standard Wasserstein-2 distance (equivalent to $\mathcal{W}_{2,1,0}$).

3.3 Error estimates

Theorem 5. *Consider the SMS-UBU scheme with friction parameter $\gamma > 0$, stepsize $h > 0$ and initial measure $\overline{\pi}_0$ and assume that $h < \frac{1}{2\gamma}$ and $\gamma \geq \sqrt{8M}$. Let the potential f be M-∇Lipschitz and of the form $f = \sum_{i=1}^{N_m} f_i$, where each f_i is m_i-strongly convex for $i = 1, ..., N_m$ and M/N_m-∇Lipschitz with minimizer $x^* \in \mathbb{R}^d$. Consider the measure of the position k-th SMS-UBU, denoted by $\tilde{\pi}_k$, which approximates the target measure π for a potential f that is M_1-Hessian Lipschitz. We have that*

$$\mathcal{W}_2(\tilde{\pi}_k, \pi) \leq \sqrt{2} \exp\left(-\frac{mh}{8\gamma}\left\lfloor\frac{k}{N_m}\right\rfloor\right)\mathcal{W}_{2,a,b}(\overline{\pi}_0, \overline{\pi}) + C(\gamma, m, M, M_1, N_m)h^2 d,$$

and if f is M_1^s-strongly Hessian Lipschitz we have that

$$\mathcal{W}_2(\tilde{\pi}_k, \pi) \leq \sqrt{2} \exp\left(-\frac{mh}{8\gamma}\left\lfloor\frac{k}{N_m}\right\rfloor\right)\mathcal{W}_{2,a,b}(\overline{\pi}_0, \overline{\pi}) + C(\gamma, m, M, M_1^s, N_m)h^2\sqrt{d}.$$

If we impose the stronger stepsize restriction $h < 1/(12\gamma N_m)$, then our bounds simplify to the form

$$\mathcal{W}_2(\tilde{\pi}_k, \pi) \leq \sqrt{2} \exp\left(-\frac{mh}{8\gamma}\left\lfloor\frac{k}{N_m}\right\rfloor\right)\mathcal{W}_{2,a,b}(\overline{\pi}_0, \overline{\pi}) + C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s)h^2 N_m^{5/2}\sqrt{d},$$

where $\tilde{\gamma} = \gamma/\sqrt{N_m}, \tilde{m} = m/N_m, \tilde{M} = M/N_m$ and $\tilde{M}_1^s = M_1^s/N_m$, and similarly when we don't assume the potential is strongly Hessian Lipschitz.

Our next result focuses on the large stepsize regime, which has also recently been considered in Shaw and Whalley (2025).

Theorem 6. *Considering the SMS-UBU scheme with friction parameter $\gamma > 0$, stepsize $h > 0$, initial measure $\overline{\pi}_0$ and assuming that $h < \frac{1}{2\gamma}$, $\gamma \geq \sqrt{8M}$ and the stochastic gradient satisfies Assumptions 2 with constants C_{SG}. Let the potential f be M-∇Lipschitz, m-strongly convex and of the form $f = \sum_{i=1}^{N_m} f_i$, where each f_i is m_i-strongly convex, where $m = \sum_{i=1}^{N_m} m_i$. Then at the k-th iteration for its measure π^k we have the non-asymptotic bound*

$$\mathcal{W}_{2,a,b}(\pi^k, \overline{\pi}) \leq \exp\left(-\frac{mh}{8\gamma}\left\lfloor\frac{k}{N_m}\right\rfloor\right)\mathcal{W}_{2,a,b}(\overline{\pi}_0, \overline{\pi}) + C(\gamma/\sqrt{N_m}, m/N_m, M/N_m)\left[\frac{C_{SG}\sqrt{h}}{N_m^{1/4}} + h\right]\sqrt{d}.$$

Remark 5. When using random permutations we have an unbiased estimator of the force at each step and we can achieve similar bounds as Gouraud et al. (2025) and Dalalyan and Karagulyan (2019) which allows us to improve upon our bounds in the large N_m, and large stepsize setting. This is the result of Theorem 6. We can demonstrate a corresponding result in the simpler setting with vanilla stochastic gradients for UBU (see Theorem 15).

Remark 6. These results can be extended to the non-convex setting using the results of Schuh (2024) or Schuh and Whalley (2024) under appropriate assumptions (convexity outside a ball), using contraction results of the continuous or discrete dynamics. The dimension and stepsize dependence would be as in the convex case, with additional dependence on the radius of the ball of non-convexity.

3.4 Numerical evaluation of bias of integrators

To illustrate the $O(h^2)$ bias for SMS-UBU, we are going to first consider a simple one dimensional Gaussian target example of the form $U(x) = U_1(x) + U_2(x)$, where $U_i(x) = \frac{(x-x_i)^2}{\sigma_i^2}$ are Gaussian potentials. We have chosen the numerical values $x_1 = -1$, $x_2 = 1$, $\sigma_1 = 0.5$, $\sigma_2 = 2$. In this example, there are only two batches when using batch size 1. Since the target is itself a Gaussian, by taking samples from the Markov chain, and the Gaussian target, we can evaluate the Wasserstein distances between them to good accuracy (it is well known that the Wasserstein distance of two one dimensional empirical distributions with the same N samples equals $\frac{1}{N}\sum_{i=1}^{N}|x_{(i)} - y_{(i)}|$). Using step sizes $h = 2^{-k}$ for $k = 2, 3, 4, 5$, and taking a large number of samples ($N = 10^7 \cdot 2^k$), we evaluated the Wasserstein bias between the Gaussian target and the stationary distribution of 9 samplers. SMS-UBU, SG-UBU, and SG-UBU with batches sampled without replacement, and the same 3 variants of BAOAB (Leimkuhler et al. (2024a)) and EM (SG-EM is called SG-HMC in the literature). The results are shown on Figure 3.4 (the errors in our bias estimates seem to be negligibly small as the results did not change noticeably when increasing the number of samples N).

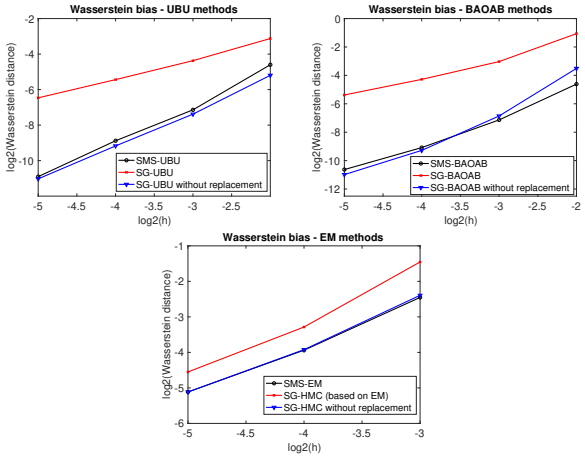

Figure 1: Wasserstein bias of stochastic integrators of kinetic Langevin dynamics for a 1D Gaussian target

These results show that SMS-UBU and SG-UBU without

replacement perform similarly, and they also slightly outperform all other methods. SMS-UBU, SG-UBU without replacement, SMS-BAOAB and SG-BAOAB without replacement appear to have a bias of $O(h^2)$ (weak order 2), while the other methods seem to have a bias of $O(h)$ (weak order 1). This is consistent with our theoretical results. SG-HMC and the other EM-based methods seem to have much larger bias than the methods based on higher order integrators (and EM based methods diverged at the largest step size $h = 2^{-2}$).

We have conducted another experiment in Bayesian multinomial logistic regression (Bishop and Nasrabadi (2006)) on the Fashion-MNIST dataset (Xiao et al. (2017)). We have used a isotropic Gaussian prior (quadratic regularizer) of the form $p(x)dx \propto \exp(-\|x\|^2/(2\sigma^2))dx$ with $\sigma = 50^{-1/2}$. The potential is strongly convex and log-concave. Using the L-BFGS algorithm, we found the global minimizer x^*, and initiated sampling algorithms from this point. We have considered five sampling algorithms: SG-HMC based on Euler-Mayurama discretization of kinetic Langevin (Chen et al. (2014)), as well as SG-BAOAB, SMS-BAOAB, SG-UBU, and SMS-UBU. Batch sizes were chosen as 200 (out of training size 60000), and we have implemented variance reduction (8) with respect to x^*. As test functions, we have used the posterior probabilities on the correct class on the test dataset (10000 images, so 10000 probabilities). To evaluate the bias accurately, we have constructed synchronous couplings of chains at stepsizes h and $h/2$ (see Section D.1 in the Appendix). One could use alternative splitting schemes, such as ABOBA as mentioned in Chen et al. (2015), but we would expect it to have worse performance compared to the SG-BAOAB integrator, since BAOAB is known to have improved bias compared to all other A,B and O splittings.

In addition to the biases, we have also evaluated the accuracy of the estimators based on posterior mean predictive probabilities of each class, and also evaluated calibration performance in terms of average negative log-likelihood on test set (NLL), adaptive calibration error (ACE) Nixon et al. (2019), and Ranked Probability Score (RPS) Constantinou and Fenton (2012).

As Figure 3.4 shows, SMS-UBU outperforms alternative methods in terms of calibration performance and accuracy at the largest stepsizes, and it has a significantly smaller bias compared to the BAOAB and EM integrators. The plots agree with our theoretical results, showing the bias to be $O(\sqrt{d}h^2)$. The dimensional dependency seems even better for these test functions, in line with the results in Chen et al. (2024).

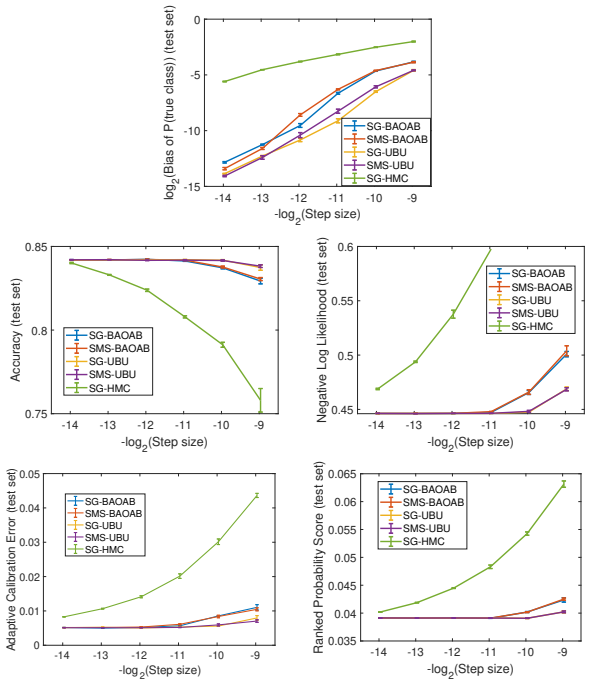

Figure 2: Top: average bias of probability of true class on test dataset for five integrators as a function of stepsize. The other plots show the effect of stepsize on accuracy and calibration performance (NLL, ACE, and RPS).

4 Bayesian uncertainty quantification in neural networks via sampling

In this section, we explain the use of our sampling methods for exploring the posterior distributions of the parameters of neural networks. The theory is based on a unimodality assumption (log-concave target distribution) whose potential has a bounded second derivative. It is well known that the loss functions of neural networks are highly multimodal and can have minimizers in areas of high curvature, i.e. the norm of the Hessian may be large. Sampling from such multimodal distributions is challenging. Mixing can be slow even if full gradients are used (Izmailov et al. (2021)).

There has been a significant amount of effort in the literature to find minimizers that lie in flat regions with low curvature, see Baldassi et al. (2020) and Foret et al. (2021). In the experiments of this paper, we have used a slowly decreasing stepsize combined with Stochastic Weight Averaging (SWA) (Izmailov et al. (2018)) to find flat minimizers. We can assess the flatness of an area near a point via the norm of the Hessian of the log-posterior at the point. Figure 7 in Section D.2 of the Appendix shows the typical norm of the Hessian of log-posterior for a CNN-based neural network trained for classification on the Fashion-MNIST dataset. The Hessian is much better behaved in the relatively smooth area in the neighbourhood of the SWA network optima.

Let x^* the SWA weights found by the algorithm. Instead of exploring the multimodal target $\pi(dx) \propto \exp(-f(x))dx$, we propose the localized distribution $\pi^*(x)$ and potential energy $f^*(x)$ defined by

$$\pi^*(dx) \propto \exp(-f^*(x))dx \text{ for } \|x - x^*\|_\infty < \rho_{\max},$$
$$f^*(x) = f(x) + \frac{1}{2\rho^2}\|x - x^*\|^2,$$

where $\rho, \rho_{\max} > 0$ are so-called localization parameters. $f^*(x)$ will be strongly convex due to inclusion of the quadratic regularizer term $\frac{1}{2\rho^2}\|x - x^*\|^2$ (for sufficiently small ρ). We further restrict the parameters to the hypercube $\Omega = \{x : \|x - x^*\|_\infty < \rho_{\max}\}$ which ensures that the norm of the Hessian $\|\nabla^2(f^*(x))\|$ is relatively small within the domain Ω (see Figure 7). Although Algorithm 2 was stated for an unconstrained domain, it is straightforward to adapt it to the hypercube Ω by implementing elastic bounces independently in each component whenever the x component exits the domain (recent theoretical results for numerical integrators on constrained spaces suggest that such bounces do not change the order of accuracy, see Leimkuhler et al. (2024b)). We chose $\rho_\infty = 6\rho$ in our experiments, which ensured that bounces rarely happened. Figure 3 shows the average training loss function for four independent SMS-UBU paths over 40 epochs started from x^* plus a multivariate Gaussian with standard deviation $\rho = 50^{-1/2}$. We have repeated this experiment 16 times, and plotted the $\hat{R}$ values (Gelman-Rubin diagnostics, computed after 10 epochs of burn-in). These plots indicate that SMS-UBU is able to approximate the localized distribution π^* efficiently, in contrast to earlier experiments exploring the whole target π (Izmailov et al. (2021)). The additional computational cost for sampling (40 epochs) is only twice the cost of optimization (20 epochs).

The local approximation π^* can still be far away from the original multimodal distribution π. To obtain a better approximation, we can use ensembles of independently obtained SWA points $x_1^*, \ldots, x_N^*$, and run N independent SMS-UBU chains initiated from these points. The full details of our approach are stated in Algorithm 3.

5 Experiments

In this section, we evaluate the accuracy and calibration performance of ensembles of deep Bayesian neural networks on three datasets: Fashion-MNIST (Xiao et al. (2017)), Celeb-A (Yang et al. (2015)), and chest X-ray (Kermany et al. (2018)). In the Celeb-A, we considered classification between blonde and brown hair colours. The networks had six convolutional layers combined with batch normalization and max pooling layers, followed by some MLP layers (these were chosen as low-rank to reduce the memory use). The Pytorch code of the them is included in Section D of the Appendix. We used a single RTX 4090 GPU for training.

Algorithm 3 Ensemble SMS-UBU with SWA centred local approximation

Require: Stepsize $h > 0$, friction parameter $\gamma > 0$. Number of sampling steps K, number of training epochs T_{train}, number of SWA epochs T_{SWA}. Localization parameters ρ, $\rho_{\max}$.
for $n = 1, 2, ..., N$ **do**

1. Randomly initialize the network weights.
2. Train network for T_{train} epochs at decreasing stepsize.
3. Continue training network with a fixed stepsize for T_{SWA} epochs, and accumulate the average weights over this period in variable $x^*_{(n)}$.
4. Obtain K samples from the distribution $\pi^*_{(n)}(x) \propto \exp\left(-f(x) - \frac{\|x - x^*_{(n)}\|^2}{2\rho^2}\right)$ using SMS-UBU initiated at $x_0 = x^*_{(n)}$, $v_0 \sim \mathcal{N}(0_d, I_d)$, with elastic bounces when exiting hypercube $\Omega_{(n)} = \{x : \|x - x^*_{(n)}\|_\infty < \rho_{\max}\}$.

end for
Output: Samples $(x_k^{(n)})_{k \in 1:K, n \in 1:N}$.

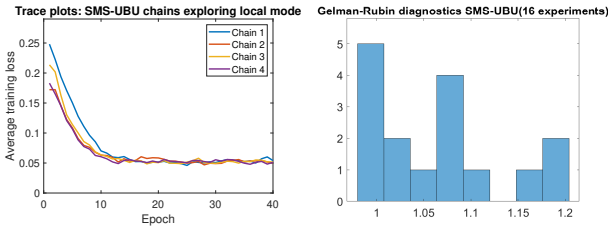

Figure 3: Left: trace plots for 4 SMS-UBU chains initialized at Gaussian perturbations of SWA weights. Right: Gelman-Rubin diagnostic $\hat{R}$ of average training loss computed for 16 SWA weights x^* obtained independently (four parallel chains each). The four chains converge to the same level, and $\hat{R}$ values are close to unity, indicating excellent mixing.

The basic setup in all experiments was the same: 15 epochs of initial training with Adam at a step polynomially decaying stepsize schedule with power equal to 1, followed by five epochs of SWA, and 40 epochs of SMS-UBU (10 epoch was discarded as burn-in, and 30 epochs were used as samples). See Section F.1 for further details.

We performed 64 independent runs, and compared the performance of ensemble methods (ensembles after 15 epochs of training, ensembles of SWA networks, and ensembles of Bayesian samples according to Algorithm 3). The results are shown in Figures 4, 5, and 6. Standard deviations were computed based on the independent runs (for example, by pooling the runs into ensembles of size four, there were 16 independent such ensembles, etc.) Bayesian networks offer small gains in accuracy over ensembles of the same size, and significant gains in calibration performance (NLL, ACE, and RPS scores). The Bayesian approach based on SMS-UBU seems to be able to provide a better modelling of uncertainty in the weights, which seems to persists even with the local π^* used in Algorithm 3.

On Figure 4, we have also included experiments with four additional integrators: SG-UBU, SG-UBU with sampling without replacement, SG-HMC (Chen et al. (2014)) and cyclical SG-HMC (Zhang et al. (2020)). As proposed in the literature, SG-HMC and cyclical SG-HMC do not use local approximation π^*, but aim to explore the whole target (using same initialization and number of epochs for all methods). These do not perform as well the UBU-based methods relying on local approximations.

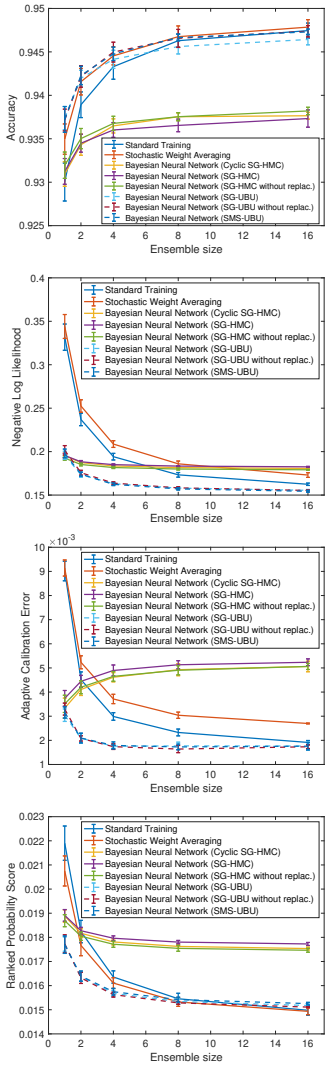

Figure 4: Accuracy and calibration results for a CNN-based network on Fashion-MNIST.

6 Conclusion

We have proposed a new sampling algorithm for Bayesian neural network posteriors, based on a kinetic Langevin integrator combined with symmetric minibatching. We demonstrated a number of important results related to convergence of the numerical scheme, in terms of Wasserstein contraction, and also in terms of the derived estimates for weak error, establishing the accuracy to be $\mathcal{O}(h^2)$. Several numerical examples demonstrate the performance of SMS-UBU against other stochastic gradient schemes. A comparison was provided to both stochastic weight averaging and standard training (optimization). We obtained substantial improvements in calibration error. In terms of future work, one possibility is to explore unbiased sampling methods for BNN posteriors (see Chada et al. (2023) and Giles et al. (2020)), which could provide an alternative to Metropolized methods such as Cobb and Jalaian (2021). A second direction could be to combine the mode connectivity framework Garipov et al. (2018); Frankle et al. (2021); Entezari et al. (2022) with SMS-UBU, potentially improving efficiency.

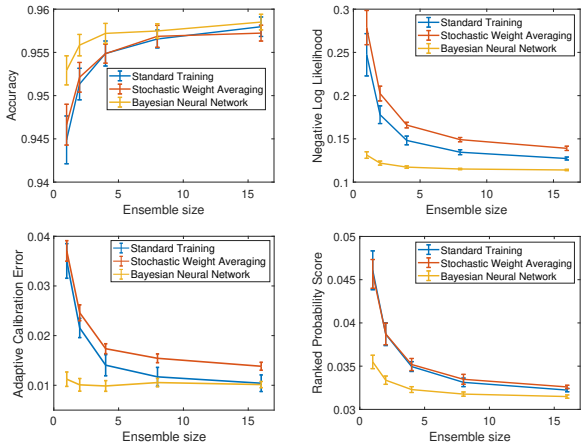

Figure 5: Accuracy and calibration results for a CNN-based network for classifying brown/blonde hair colour on the Celeb-A dataset.

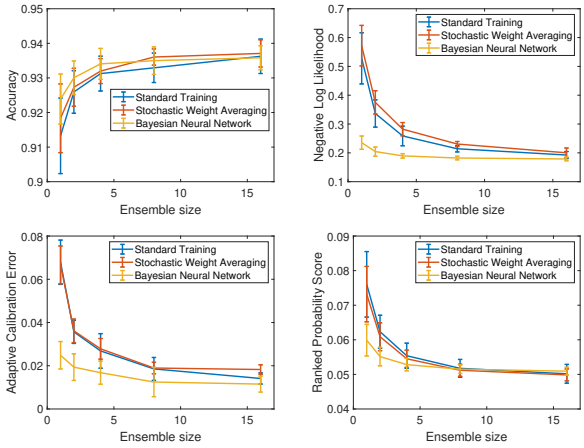

Figure 6: Accuracy and calibration results for a CNN-based network for detecting pneumonia on a chest X-ray dataset.

Acknowledgements

D.P. and P.A.W. contributed equally to this work.

References

T. Abe, E. K. Buchanan, G. Pleiss, R. Zemel, and J. P. Cunningham. Deep ensembles work, but are they necessary? *Advances in Neural Information Processing Systems*, 35: 33646–33660, 2022.

Z. Alfonso Álamo. *Word Series for the Numerical Integration of Stochastic Differential Equations*. PhD thesis, Universidad de Valladolid, 2021.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

J. Baker, P. Fearnhead, E. B. Fox, and C. Nemeth. Control variates for stochastic gradient MCMC. *Stat. Comput.*, 29(3):599–615, 2019. ISSN 0960-3174,1573-1375. doi: 10.1007/s11222-018-9826-2. URL https://doi.org/10.1007/s11222-018-9826-2.

C. Baldassi, F. Pittorino, and R. Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020.

N. Band, T. G. J. Rudner, Q. Feng, and et al. Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks. *Neural Information Processing Systems (NeurIPS) 2021 Datasets and Benchmarks Track Proceedings*, 2021.

R. Bardenet, A. Doucet, and C. Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(47):1–43, 2017.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

N. Bou-Rabee and H. Owhadi. Long-run accuracy of variational integrators in the stochastic context. *SIAM J. Numer. Anal.*, 48(1):278–297, 2010. ISSN 0036-1429,1095-7170. doi: 10.1137/090758842. URL https://doi.org/10.1137/090758842.

G. Bussi and M. Parrinello. Accurate sampling using Langevin dynamics. *Phys. Rev. E*, 75: 056707, May 2007. doi: 10.1103/PhysRevE.75.056707. URL https://link.aps.org/doi/10.1103/PhysRevE.75.056707.

N. K. Chada, B. Leimkuhler, D. Paulin, and P. A. Whalley. Unbiased kinetic Langevin Monte Carlo with inexact gradients. *arXiv preprint arXiv:2311.05025*, 2023.

M. Chak and P. Monmarché. Reflection coupling for unadjusted generalized Hamiltonian Monte Carlo in the nonconvex stochastic gradient case. *arXiv preprint arXiv:2310.18774*, 2023.

N. Chatterji, N. Flammarion, Y. Ma, P. Bartlett, and M. Jordan. On the theory of variance reduction for stochastic gradient Monte Carlo. In *International Conference on Machine Learning*, pages 764–773. PMLR, 2018.

C. Chen, N. Ding, and L. Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. *Neural Information Processing Systems (NeurIPS) 2015 Proceedings*, 2015.

T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. *In International Conference on Machine Learning*, page 1683–1691, 2014.

Y. Chen and K. Gatmiry. When does Metropolized Hamiltonian Monte Carlo provably outperform Metropolis-adjusted Langevin algorithm? *arXiv preprint arXiv:2304.04724*, 2023.

Y. Chen, X. Cheng, J. Niles-Weed, and J. Weare. Convergence of Unadjusted Langevin in High Dimensions: Delocalization of Bias. *arXiv preprint arXiv:2408.13115*, 2024.

X. Cheng, N. S. Chatterji, P. L. Bartlett, and M. I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Conference on learning theory*, pages 300–323. PMLR, 2018.

A. D. Cobb and B. Jalaian. Scaling Hamiltonian Monte Carlo inference for Bayesian neural networks with symmetric splitting. *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, 161: 675–685, 2021.

A. C. Constantinou and N. E. Fenton. Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models. *Journal of Quantitative Analysis in Sports*, 8(1), 2012.

M. Cranmer, D. Tamayo, H. Rein, P. Battaglia, S. Hadden, P. J. Armitage, S. Ho, and D. N. Spergel. A Bayesian neural network predicts the dissolution of compact planetary systems. *PNAS*, 118(40), 2021.

A. S. Dalalyan and A. Karagulyan. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *Stochastic Process. Appl.*, 129(12):5278–5311, 2019. ISSN 0304-4149,1879-209X. doi: 10.1016/j.spa.2019.02.016. URL https://doi.org/10.1016/j.spa.2019.02.016.

A. S. Dalalyan and L. Riou-Durand. On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli*, 26(3):1956–1988, 2020. ISSN 1350-7265,1573-9759. doi: 10.3150/19-BEJ1178.

G. Detommaso, A. Gasparin, and M. e. a. Donini. Fortuna: A library for uncertainty quantification in deep learning. *J. Mach. Learn. Res.*, 25, 2023.

M. Dusenberry, G. Jerfel, Y. Wen, Y. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, and D. Tran. Efficient and scalable Bayesian neural nets with rank-1 factors. *In*

International Conference on Machine Learning, pages 2782–2792, 2020.

A. Eberle, A. Guillin, and R. Zimmer. Couplings and quantitative contraction rates for Langevin dynamics. *Ann. Probab.*, 47(4):1982–2010, 2019. ISSN 0091-1798,2168-894X. doi: 10.1214/18-AOP1299.

R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *Conference on Learning Theory*. PMLR, 2022.

P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

J. Frankle, G. Karolina Dziugaite, and D. e. a. Roy. Linear mode connectivity and the lottery ticket hypothesis. *In International Conference on Machine Learning*, pages 3259–3269, 2021.

G. Franzese, D. Milios, and M. e. a. Filippone. Revisiting the effects of stochasticity for Hamiltonian samplers. *In International Conference on Machine Learning*, pages 6744–6778, 2022.

T. Garipov, P. Izmailov, and D. e. a. Podoprikhin. Loss surfaces, mode connectivity, and fast ensembling of DNNs. *Neural Information Processing Systems (NeurIPS) 2018 Proceedings*, 2018.

N. Geneva and N. Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403(109056), 2020.

M. B. Giles, M. B. Majka, L. Szpruch, S. J. Vollmer, and K. C. Zygalakis. Multi-level Monte Carlo methods for the approximation of invariant measures of stochastic differential equations. *Stat. Comput.*, 30(3): 507–524, 2020. ISSN 0960-3174,1573-1375. doi: 10.1007/s11222-019-09890-0. URL https://doi.org/10.1007/s11222-019-09890-0.

N. Gouraud, P. L. Bris, A. Majka, and P. Monmarché. Hmc and underdamped langevin united in the unadjusted convex smooth case. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1):278–303, 2025. doi: 10.1137/23M1608963. URL https://doi.org/10.1137/23M1608963.

A. M. Horowitz. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2):247–252, 1991.

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilson. Averaging weights leads to wider optima and better generalization. 34th Conference on Uncertainty in Artificial Intelligence 2018, pages 876–885, 2018.

P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. Wilson. What are Bayesian neural network posteriors really like? *In Proceedings of the 38th International Conference on Machine Learning*, 139:4629–4640, 2021.

D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5): 1122–1131, 2018.

M. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava. Fast and scalable Bayesian deep learning by weight-perturbation in ADAM. *In International Conference on Machine Learning*, 161:675–685, 2021.

D.-M. Koh, N. Papanikolaou, U. Bick, and et al. Artificial intelligence and machine learning in cancer imaging. *Commun Med*, 2022. doi: 10.1038/s43856-022-00199-0.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

B. Leimkuhler and C. Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Appl. Math. Res. Express. AMRX*, (1):34–56, 2013. ISSN 1687-1200,1687-1197. doi: 10.1093/amrx/abs010. URL https://doi.org/10.1093/amrx/abs010.

B. Leimkuhler, C. Matthews, and G. Stoltz. The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. *IMA J. Numer. Anal.*, 36(1):13–79, 2016. ISSN 0272-4979,1464-3642. doi: 10.1093/imanum/dru056. URL https://doi.org/10.1093/imanum/dru056.

B. Leimkuhler, D. Paulin, and P. A. Whalley. Contraction rate estimates of stochastic gradient kinetic Langevin integrators*. *ESAIM: M2AN*, 58(6):2255–2286, 2024a. doi: 10.1051/m2an/2024038. URL https://doi.org/10.1051/m2an/2024038.

B. Leimkuhler, A. Sharma, and M. V. Tretyakov. Numerical integrators for confined Langevin dynamics. *arXiv preprint arXiv:2404.16584*, 2024b.

B. J. Leimkuhler, D. Paulin, and P. A. Whalley. Contraction and convergence rates for discretized kinetic Langevin dynamics. *SIAM J. Numer. Anal.*, 62(3): 1226–1258, 2024c. ISSN 0036-1429,1095-7170. doi: 10.1137/23M1556289. URL https://doi.org/10.1137/23M1556289.

J. Levinson and et al. Towards fully autonomous driving: Systems and algorithms. *Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011.

D. J. C. MacKay. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505, 1995.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncer-

tainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

R. Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):58–77, 1929.

P. Monmarché. High-dimensional MCMC with a standard splitting scheme for the underdamped Langevin diffusion. *Electron. J. Stat.*, 15(2):4117–4166, 2021. ISSN 1935-7524. doi: 10.1214/21-ejs1888. URL https://doi.org/10.1214/21-ejs1888.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer Science & Business Media, 2012.

C. Nemeth and P. Fearnhead. Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran. Measuring calibration in deep learning. In *CVPR workshops*, volume 2, 2019.

T. Papamarkou, J. Hinkle, T. Young, and D. Womble. Challenges in Markov chain Monte Carlo for Bayesian neural networks. *Statistical Science*, 37(3):435–442, 2022.

D. Paulin and P. A. Whalley. Correction to "Wasserstein distance estimates for the distributions of numerical approximations to ergodic stochastic differential equations". *Journal of Machine Learning Research*, 25(376): 1–9, 2024.

G. A. Pavliotis. Stochastic processes and applications. *Texts in applied mathematics*, 60, 2014.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

J. M. Sanz-Serna and K. C. Zygalakis. Wasserstein distance estimates for the distributions of numerical approximations to ergodic stochastic differential equations. *J. Mach. Learn. Res.*, 22:242–1, 2021.

K. Schuh. Global contractivity for Langevin dynamics with distribution-dependent forces and uniform in time propagation of chaos. *Ann. Inst. Henri Poincaré Probab. Stat.*, 2024.

K. Schuh and P. A. Whalley. Convergence of kinetic langevin samplers for non-convex potentials. *arXiv preprint arXiv:2405.09992*, 2024.

I. Sekkat and G. Stoltz. Quantifying the mini-batching error in Bayesian inference for adaptive Langevin dynamics. *J. Mach. Learn. Res.*, 24:Paper No. [329], 58, 2023. ISSN 1532-4435,1533-7928.

L. Shaw and P. A. Whalley. Random reshuffling for stochastic gradient Langevin dynamics. *arXiv preprint arXiv:2501.16055*, 2025.

L. Shen, Y. Sun, Z. Yu, D. Liang, X. Tian, and D. Tao. On Efficient Training of Large-Scale Deep Learning Models: A Literature Review. *arXiv preprint arxiv:2304.03589*, 2023.

H. Thadeshwar, V. Shah, M. Jain, R. Chaudhari, and V. Badgujar. Artificial intelligence based self-driving car. *: 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020. doi: 10.1109/ICCCSP49186.2020.9315223.

S. J. Vollmer, K. C. Zygalakis, and Y. W. Teh. Exploration of the (non-)asymptotic bias and variance of stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.*, 17: Paper No. 159, 45, 2016. ISSN 1532-4435,1533-7928.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688, 2011.

P. A. Whalley. *Kinetic Langevin Monte Carlo methods*. PhD thesis, The University of Edinburgh, 2024.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE international conference on computer vision*, pages 3676–3684, 2015.

R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. *In International Conference on Learning Representations*, 2020.

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
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. No

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes

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

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes

(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

(d) A description of the computing infrastructure used. Yes

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

(b) The license information of the assets, if applicable. Not Applicable

(c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable

(d) Information about consent from data providers/curators. Not Applicable

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. 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. Not Applicable

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Sampling from Bayesian Neural Network Posteriors with Symmetric Minibatch Splitting Langevin Dynamics: Supplementary Materials

A Algorithms

Algorithm 4 Stochastic Gradient Euler-Maruyama (SG-HMC)

- Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$ and friction parameter $\gamma > 0$.
- for $k = 1, 2, ..., K$ do

  Sample $\omega_k \sim \rho$
  Sample $\xi_k \sim \mathcal{N}(0_d, I_d)$
  $x_k \to x_{k-1} + hv_{k-1}$
  $v_k \to v_{k-1} - h\mathcal{G}(x_{k-1}, \omega_k) - h\gamma v_{k-1} + \sqrt{2\gamma h}\xi_k$
- Output: Samples $(x_k)_{k=0}^K$.

Algorithm 5 Stochastic Gradient BAOAB (SG-BAOAB)

- Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$ and friction parameter $\gamma > 0$.
- Sample $\omega_1 \sim \rho$
- $G_0 \to \mathcal{G}(x_0, \omega_1)$
- for $k = 1, 2, ..., K$ do
  - (B) $v \to v_{k-1} - \frac{h}{2} G_{k-1}$
  - (A) $x \to x_{k-1} + \frac{h}{2} v$
    Sample $\xi_k \sim \mathcal{N}(0_d, I_d)$
  - (O) $v \to \eta v + \sqrt{1-\eta^2}\xi_k$
  - (A) $x_k \to x + \frac{h}{2} v$
    Sample $\omega_{k+1} \sim \rho$
    $G_k \to \mathcal{G}(x_k, \omega_{k+1})$
  - (B) $v_k \to v - \frac{h}{2} G_k$
- Output: Samples $(x_k)_{k=0}^K$.

B Convergence of the UBU scheme

Proposition 7 (Proposition C.6 of Chada et al. (2023)). *Suppose that f satisfies Assumptions 3 and 4. Let $a = \frac{1}{M}$, $b = \frac{1}{\gamma}$, $c(h) = \frac{mh}{8\gamma}$ and P_h denote the transition kernel for a step of UBU with stepsize $h > 0$. For all $\gamma \geq \sqrt{8M}$, $h < \frac{1}{2\gamma}$, $1 \leq p \leq \infty$, $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^{2d})$ and for all $n \in \mathbb{N}$,*

$$\mathcal{W}_{p,a,b}\left(\nu P_h^n, \mu P_h^n\right) \leq (1 - c(h))^n \, \mathcal{W}_{p,a,b}\left(\nu, \mu\right).$$

Further to this, P_h has a unique invariant measure π_h satisfying $\pi_h \in \mathcal{P}_p(\mathbb{R}^{2d})$ for all $1 \leq p \leq \infty$.

Algorithm 6 Symmetric Minibatch Splitting BAOAB (SMS-BAOAB)

Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$, friction parameter $\gamma > 0$ and number of minibatches N_m.
for $i = 1, 2, ..., \lceil K/2N_m \rceil$ **do**
  Sample $\omega_1, ..., \omega_{N_m} \in [N_D]^{N_b}$ uniformly without replacement.
  Define $\omega_{N_m+1} := \omega_1$.
  if i=1 **then**
    $G_0 \to \mathcal{G}(x_0, \omega_1)$.
  else
    $G_0 \to G_{N_m}$.
  end if
  Forward Sweep

  for $k = 1, 2, ..., \min\{N_m, K - (2i-2)N_m\}$ **do**
(B) $v \to v_{(2i-2)N_m+k-1} - \frac{h}{2} G_{k-1}$
(A) $x \to x_{(2i-2)N_m+k-1} + \frac{h}{2} v$
  Sample $\xi_k \sim \mathcal{N}(0_d, I_d)$
(O) $v \to \eta v + \sqrt{1-\eta^2}\xi_k$
(A) $x_{(2i-2)N_m+k} \to x + \frac{h}{2} v$
  $G_k \to \mathcal{G}(x_k, \omega_{k+1})$
(B) $v_{(2i-2)N_m+k} \to v - \frac{h}{2} G_k$
  end for

  Backward Sweep

  for $k = 1, 2, ..., \min\{N_m, K - (2i-1)N_m\}$ **do**
(B) $v \to v_{(2i-1)N_m+k-1} - \frac{h}{2} G_{N_m+1-k}$
(A) $x \to x_{(2i-1)N_m+k-1} + \frac{h}{2} v$
  Sample $\xi_k \sim \mathcal{N}(0_d, I_d)$
(O) $v \to \eta v + \sqrt{1-\eta^2}\xi_k$
(A) $x_{(2i-1)N_m+k} \to x + \frac{h}{2} v$
  $G_k \to \mathcal{G}(x_k, \omega_{N_m+1-k})$
(B) $v_{(2i-1)N_m+k} \to v - \frac{h}{2} G_k$
  end for
end for
Output: Samples $(x_k)_{k=0}^K$.

C Wasserstein bias bounds

We use the following formulation of kinetic Langevin dynamics in the analysis of the UBU scheme in the full gradient setting (as in Sanz-Serna and Zygalakis (2021)) and alternative schemes with the inexact gradient splitting methods.

It is derived via Itô's formula on the product $e^{\gamma t} V_t$, we then have for initial condition $(X_0, V_0) \in \mathbb{R}^{2d}$:

$$V_t = \mathcal{E}(t) V_0 - \int_0^t \mathcal{E}(t-s) \nabla f(X_s) ds + \sqrt{2\gamma} \int_0^t \mathcal{E}(t-s) dW_s, \tag{13}$$

$$X_t = X_0 + \mathcal{F}(t) V_0 - \int_0^t \mathcal{F}(t-s) \nabla f(X_s) ds + \sqrt{2\gamma} \int_0^t \mathcal{F}(t-s) dW_s, \tag{14}$$

where

$$\mathcal{E}(t) = e^{-\gamma t} \qquad \mathcal{F}(t) = \frac{1 - e^{-\gamma t}}{\gamma}. \tag{15}$$

Then the UBU scheme (as in Sanz-Serna and Zygalakis (2021)) can be expressed as

$$v_{k+1} = \mathcal{E}(h)v_k - h\mathcal{E}(h/2)\nabla f(y_k) + \sqrt{2\gamma}\int_{kh}^{(k+1)h} \mathcal{E}((k+1)h-s)dW_s, \tag{16}$$

$$y_k = x_k + \mathcal{F}(h/2)v_k + \sqrt{2\gamma}\int_{kh}^{(k+1/2)h} \mathcal{F}((k+1/2)h-s)dW_s, \tag{17}$$

$$x_{k+1} = x_k + \mathcal{F}(h)v_k - h\mathcal{F}(h/2)\nabla f(y_k) + \sqrt{2\gamma}\int_{kh}^{(k+1)h} \mathcal{F}((k+1)h-s)dW_s, \tag{18}$$

for comparison with the true dynamics via (13) and (14).

We now define the stochastic gradient scheme (see Definition 1) by

$$\tilde{v}_{k+1} = \mathcal{E}(h)\tilde{v}_k - h\mathcal{E}(h/2)\mathcal{G}(\tilde{y}_k, \omega_{k+1}) + \sqrt{2\gamma}\int_{kh}^{(k+1)h} \mathcal{E}((k+1)h-s)dW_s, \tag{19}$$

$$\tilde{y}_k = \tilde{x}_k + \mathcal{F}(h/2)\tilde{v}_k + \sqrt{2\gamma}\int_{kh}^{(k+1/2)h} \mathcal{F}((k+1/2)h-s)dW_s, \tag{20}$$

$$\tilde{x}_{k+1} = \tilde{x}_k + \mathcal{F}(h)\tilde{v}_k - h\mathcal{F}(h/2)\mathcal{G}(\tilde{y}_k, \omega_{k+1}) + \sqrt{2\gamma}\int_{kh}^{(k+1)h} \mathcal{F}((k+1)h-s)dW_s. \tag{21}$$

First, considering the full gradient scheme, we have the following L^2 error bound to the true diffusion.

Proposition 8. *Assume that $h < 1/2\gamma$ and $\gamma \geq \sqrt{M}$ and consider the kinetic Langevin dynamics and the UBU scheme (with full gradients and iterates $(x_n, v_n)_{n\in\mathbb{N}}$) with synchronously coupled Brownian motion initialized from the target measure $(x_0, v_0) = (X_0, V_0) \sim \pi$. Suppose that Assumption 4 and Assumption 5 are satisfied, then for $k \in \mathbb{N}$ we have that for $(\Delta_x^k, \Delta_v^k) := (x_k - X_{kh}, v_k - V_{kh})$*

$$\left\|(\Delta_x^k, \Delta_v^k)\right\|_{L^2,a,b} \leq \frac{3\gamma^2}{M} e^{3\max\{\gamma, \frac{2M}{\gamma}\}hk}\left(\mathbf{C} + \frac{5h^3}{48}(k+1)\left(5M + \gamma M^{1/2}\right)d^{1/2}\right),$$

where

$$\mathbf{C} = \gamma^{-1}(k+1)\frac{h^3\sqrt{d}}{24}\left(3M_1\sqrt{d} + M^{3/2} + \gamma M\right) + \sqrt{2\gamma^{-1}}\sqrt{\frac{(k+1)h^5M^2d}{192}},$$

and if Assumption 6 is satisfied this is refined to

$$\mathbf{C} = \gamma^{-1}(k+1)\frac{h^3\sqrt{d}}{24}\left(3M_1^s + M^{3/2} + \gamma M\right) + \sqrt{2\gamma^{-1}}\sqrt{\frac{(k+1)h^5M^2d}{192}}.$$

Proof. The result follows using the same argument as was used to establish (Schuh and Whalley, 2024, Lemma 13) with the choice of $\alpha = 5/2$ in L^2 rather than L^1 and using the equivalence of norms. □

Lemma 9. *Considering a potential of the form $f(x) = \sum_{i=1}^{N_m} f_i(x)$, where we assume that f and each f_i for $i = 1, ..., N_m$ share the same minimizer $x^* \in \mathbb{R}^{2d}$ and each $\nabla^2 f_i \prec M/N_m$ has a M/N_m-Lipschitz gradient, where f is M-∇Lipschitz and m-strongly convex we have the following moment bound*

$$\int_{\mathbb{R}^d} \|\nabla f_i(x)\| e^{-f(x)}dx \leq \frac{M}{N_m}\sqrt{\frac{d}{m}}.$$

Proof. This result follows from (Monmarché, 2021, Lemma 30) and the fact that ∇f_i is Lipschitz with minimizer $x^* \in \mathbb{R}^d$. □

Remark 7. We do not need to make the assumption that all the f_i's have the same minimizer, but we do this in order to simplify the estimates.

Lemma 10. *Let $f : \mathbb{R}^d \to \mathbb{R}$ of the form $f = \sum_{i=1}^{N_D} f_i$ have an unbiased estimator $\mathcal{G}(\cdot, \omega) = N f_\omega$, where the random variable $\omega \in [N_D]^{N_b}$, then we consider a sequence of random variables $(\omega_i)_{i=1}^{N_m}$ which are sampled uniformly without replacement, where $N_m = N_D/N_b \in \mathbb{N}$. $(x_i)_{i=0}^{N_m}$ are a sequence of random variables which are independent of ω. In this setting, we have the estimate*

$$\mathbb{E}\left[\left\|\sum_{i=1}^{N_m}\left(\mathcal{G}(x_i,\omega_i)-f(x_i)\right)\right\|^2\right] \leq \frac{5}{2}\sum_{i=1}^{N_m}\mathbb{E}\left[\|(\mathcal{G}(x_i,\omega_i)-f(x_i))\|^2\right]. \tag{22}$$

Proof. Firstly, we expand the square and we consider the cross terms which are of the form $\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)\right]$ for $i \neq j$. Let $\omega_j' = \omega_j$ with probability $1-\frac{1}{N_m}$, and $\omega_j' = \omega_i$ with probability $\frac{1}{N_m}$. Then it is easy to see that ω_j' is uniformly distributed on the set $\{1,\dots,N_m\}$, independently of ω_i. As a consequence

$$\begin{aligned}&\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)\right] = \mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)\left(f(x_j)-\mathcal{G}(x_j,\omega_j')\right)\right]\\&+\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)\left(\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)-\left(f(x_j)-\mathcal{G}(x_j,\omega_j')\right)\right)\right],\end{aligned}$$

then the first term has zero expectation. The second term can be written as

$$\begin{aligned}&\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)\left(\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)-\left(f(x_j)-\mathcal{G}(x_j,\omega_j')\right)\right)\mathbb{1}[\omega_j \neq \omega_{j'}]\right] \leq\\&\frac{1}{2}\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)^2\mathbb{1}[\omega_j \neq \omega_{j'}]\right]\\&+\frac{1}{2}\mathbb{E}\left[\left(\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)\mathbb{1}[\omega_j \neq \omega_{j'}]-\left(f(x_j)-\mathcal{G}(x_j,\omega_j')\right)\mathbb{1}[\omega_j \neq \omega_{j'}]\right)^2\right]\\&\leq \frac{1}{2N_m}\mathbb{E}\left[\left(f(x_i)-\mathcal{G}(x_i,\omega_i)\right)^2\right]+\frac{1}{N_m}\mathbb{E}\left[\left(f(x_j)-\mathcal{G}(x_j,\omega_j)\right)^2+\left(f(x_j)-\mathcal{G}(x_j,\omega_j')\right)^2\right],\end{aligned}$$

and summing up the terms we have the required result. □

Proof of Theorem 6. We estimate the difference between the full-gradient UBU scheme and the stochastic gradient scheme UBU scheme as follows. We use the notation Δ_x^k and Δ_v^k to be the difference in position and velocity at iteration $k \in \mathbb{N}$ respectively. We also use synchronously coupled Brownian motion.

Using (16) and (19) we have

$$\begin{aligned}\Delta_v^k &= \mathcal{E}(h)^k\Delta_v^0 - h\mathcal{E}(h/2)\sum_{i=1}^{k}\mathcal{E}(h)^{k-(i-1)}\left(\nabla f(y_{i-1})-N_m\nabla f_{i+1}(\tilde{y}_{i-1})\right)\\&= \mathcal{E}(h)^k\Delta_v^0 - h\mathcal{E}(h/2)\sum_{i=0}^{\lfloor\frac{k}{2N_m}\rfloor-1}\sum_{j=0}^{2N_m-1}\mathcal{E}(h)^{k-1-2iN_m-j}\left(\nabla f(y_{2N_m i+j})-N_m\nabla f_{2iN_m+j+1}(\tilde{y}_{2iN_m+j})\right)\\&- h\mathcal{E}(h/2)\sum_{i=2N_m\lfloor k/2N_m\rfloor}^{k}\mathcal{E}(h)^{k-i}\left(\nabla f(y_{i-1})-N_m\nabla f_{i+1}(\tilde{y}_{i-1})\right),\end{aligned}$$

and therefore using the fact that the ω_i random variables are independent between each block of size $2N_m$ and applying Lemma 10 to the forward and backward sweeps individually we have the following L^2 estimate

$$\|\Delta_v^k\|_{L^2} \leq \|\Delta_v^0\|_{L^2} + \sum_{i=0}^{k-1}hM\|y_i-\tilde{y}_i\|_{L^2} + \sqrt{5}h\mathcal{E}(h/2)\sqrt{\sum_{i=0}^{k-1}\|\nabla f(y_i)-N_m\nabla f_{i+1}(y_i)\|_{L^2}^2}.$$

Considering x we similarly have

$$\|\Delta_x^k\|_{L^2} \leq \|\Delta_x^0\|_{L^2} + \sum_{i=0}^{k-1}\left[h^2M\|y_i-\tilde{y}_i\|_{L^2} + h\|\Delta_v^i\|_{L^2}\right] + \sqrt{5}h^2\sqrt{\sum_{i=0}^{k-1}\|\nabla f(y_i)-N_m\nabla f_{i+1}(y_i)\|_{L^2}^2},$$

and therefore if they are initialized at the same point we can combine the estimates and get

$$\|(\Delta_x^k, \Delta_v^k)\|_{L^2,a,0} \leq 4h\sqrt{M}\sum_{i=0}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,0} + \frac{2\sqrt{5}h}{\sqrt{M}}\sqrt{\sum_{i=0}^{k-1}\|\nabla f(y_i) - N_m\nabla f_{i+1}(y_i)\|_{L^2}^2}$$
$$\leq 4h\sqrt{M}\sum_{i=0}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,0} + 2\sqrt{5}h\sqrt{M}C_{SG}\sqrt{\sum_{i=0}^{k-1}\left\|y_i - X_{(i+1/2)h}\right\|_{L^2}^2 + k\frac{d}{m}},$$

where we have used that $\|x - x^*\|_{L^2} \leq \sqrt{d/m}$ for $x \sim \pi$. By using that $\|y_i - X_{(i+1/2)}\|_{L^2} \leq C\frac{M}{m}h\sqrt{d}$ (see (Chada et al., 2023, Proposition H.3)) we have

$$\leq 4h\sqrt{M}\sum_{i=0}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,0} + Ch\sqrt{M}C_{SG}\sqrt{k\frac{M^2}{m^2}h^2d + k\frac{d}{m}}$$
$$\leq Ch\sqrt{M}C_{SG}e^{4hk\sqrt{M}}\sqrt{kd\left(\frac{M^2}{m^2}h^2 + \frac{1}{m}\right)}.$$

Now using an interpolation argument, which is presented in detail later in the proof of Theorem 15 with blocks of size $\lfloor 1/4h\sqrt{M}\rfloor$ we have the required estimate by combining the result with (Chada et al., 2023, Proposition H.3) or Proposition 8 and using contraction of the UBU scheme at each step with different convexity constants $(m_i)_{i=1}^{N_m}$. □

Lemma 11. *Under the same conditions as Theorem 6 we have that for $i \leq 2N_m - 1$*

$$\|\tilde{\Delta}^i\|_{L^2} + h\|\tilde{\Delta}_v^i\|_{L^2} \leq \sqrt{2}\|\tilde{\Delta}^0\|_{L^2,a,b} + \mathcal{O}\left(h^2N_mM(C_{SG}+1)e^{8hN_m\gamma}\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2} + \frac{1}{m}\right)}\right).$$

Proof. Assuming that $\tilde{\Delta}^0 = 0$, we have that for all $i = 0, ..., 2N_m - 1$ (using (Chada et al., 2023, Proposition H.3) or Theorem 6) that

$$\|\tilde{\Delta}_v^i\|_{L^2} \leq e^{8hN_m\gamma}\mathcal{O}\left(hM(C_{SG}+1)\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2} + \frac{1}{m}\right)}\right)$$

and using (14), (21) and discrete Grönwall inequality we have

$$\|\tilde{\Delta}_x^i\|_{L^2} \leq \mathcal{O}\Bigg(h^2N_m\sqrt{M}C_{SG}e^{8hN_m\gamma}\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2} + \frac{1}{m}\right)}$$
$$+ \sum_{j=0}^{i-1}\int_0^h \|\mathcal{F}(h/2)N_m\nabla f_j(\tilde{y}_j) - \mathcal{F}(h-s)\nabla f(X_{s+jh})\|_{L^2}ds\Bigg)$$
$$\leq \mathcal{O}\left(h^2N_mM(C_{SG}+1)e^{8hN_m\gamma}\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2} + \frac{1}{m}\right)} + \sum_{j=0}^{i-1}h^2M(\|\tilde{\Delta}_x^j\|_{L^2} + h\|\tilde{\Delta}_v^j\|_{L^2})\right)$$
$$\leq \mathcal{O}\left(h^2N_mM(C_{SG}+1)e^{8hN_m\gamma}\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2} + \frac{1}{m}\right)}\right).$$

Then for $\tilde{\Delta}^0 \neq 0$ we can use Proposition 7 and the triangle inequality to achieve the required result.

□

Proposition 12. *Suppose we have an SMS-UBU discretization with a potential that is M-∇ Lipschitz and M_1-Hessian Lipschitz and of the form $f = \sum_{i=1}^{N_m} f_i$, where $\nabla^2 f_i \prec MI_D/N_m$ for all $i = 1, ..., N_m$, $h < \min\left\{\frac{1}{2\gamma}, \frac{1}{2\sqrt{M}}\right\}$, for $l \in \mathbb{N}$ $(\tilde{\Delta}_x^{2lN_m}, \tilde{\Delta}_v^{2lN_m}) := (\tilde{x}_{2lN_m} - X_{2lN_mh}, \tilde{v}_{2lN_m} - V_{2lN_mh})$, where the discretization and continuous dynamics have*

synchronously coupled Brownian motion and they are all initialized from the target measure, $(X_0, V_0) \sim \pi$. *Then if* f *is* M_1*-Hessian Lipschitz we have that*

$$\|(\tilde{x}_{2lN_m} - X_{2lN_m h}, \tilde{v}_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq e^{16h\gamma lN_m} C(\gamma, m, M, M_1, N_m, C_{SG}) \left(lh^3 d + h^{5/2}\sqrt{ld}\right),$$

and if f *is* M_1^s*-strongly Hessian Lipschitz*

$$\|(\tilde{x}_{2lN_m} - X_{2lN_m h}, \tilde{v}_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq e^{16h\gamma lN_m} C(\gamma, m, M, M_1^s, N_m) \left(lh^3 + h^{5/2}\sqrt{l}\right)\sqrt{d}.$$

If we impose the stronger stepsize restriction $h < 1/(2\gamma N_m)$*, then our bounds simplify to the form*

$$\|(\tilde{x}_{2lN_m} - X_{2lN_m h}, \tilde{v}_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq e^{16h\gamma lN_m} C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) \left(lh^3 N_m^4 + h^{5/2}\sqrt{l}N_m^{13/4}\right)\sqrt{d},$$

where $\tilde{\gamma} = \gamma/\sqrt{N_m}, \tilde{m} = m/N_m, \tilde{M} = M/N_m$ *and* $\tilde{M}_1^s = M_1^s/N_m$.

Remark 8. We could instead assume that $\nabla^2 f_i \prec M_i I_D$, for all $i = 1, ..., N_m$, each f_i having a different gradient Lipschitz constant, but to simplify notation in the argument we assume the stronger assumption $\nabla^2 f_i \prec M I_D/N_m$.

Proof. We consider the position and velocity components separately.

Velocity component

Firstly, we write

$$\tilde{v}_{2N_m} - V_{2N_m h} = \underbrace{\tilde{v}_{2N_m} - v_{2N_m}}_{\text{(I)}} + \underbrace{v_{2N_m} - V_{2N_m h}}_{\text{(II)}},$$

and similarly, in x, we can bound the (II) using the full gradient scheme bounds in Proposition 8. Then we consider (I), the distance to full gradient discretization. We define $(\delta_x^k, \delta_v^k) := (\tilde{x}_k - x_k, \tilde{v}_k - v_k)$, for $k = 0, ..., 2N_m$, then

$$\begin{aligned}
\delta_v^{2N_m} &= \mathcal{E}(h)\delta_v^{2N_m-1} - h\mathcal{E}(h/2)(N_m \nabla f_1(\tilde{y}_{2N_m-1}) - \nabla f(y_{2N_m-1})) \\
&= \mathcal{E}^{2N_m}(h)\delta_v^0 - h\mathcal{E}(h/2)\Bigg[\sum_{i=0}^{N_m-1} \mathcal{E}(h)^{N_m-i-1}\left(N_m \nabla f_{N_m-i}(\tilde{y}_{N_m+i}) - \nabla f(y_{N_m+i})\right) \\
&+ \sum_{i=0}^{N_m-1} \mathcal{E}(h)^{i+N_m}\left(N_m \nabla f_{N_m-i}(\tilde{y}_{N_m-i-1}) - \nabla f(y_{N_m-i-1})\right)\Bigg] \\
&= \mathcal{E}^{2N_m}(h)\delta_v^0 + (\star),
\end{aligned}$$

where without loss of generality we have assumed that they are ordered $1, ..., N_m$ and allow for random permutations of this ordering.

We now consider

$$\begin{aligned}
(\star) = -h \sum_{i=0}^{N_m-1} \Bigg[& C^i_{N_m-i} + N_m \mathcal{E}(h)^{N_m-i-1/2}\left(\nabla f_{N_m-i}(\tilde{y}_{N_m+i}) - \nabla f_{N_m-i}(X_{(N_m+i+1/2)h})\right) \\
& + N_m \mathcal{E}(h)^{N_m+i+1/2}\left(\nabla f_{N_m-i}(\tilde{y}_{N_m-i-1}) - \nabla f_{N_m-i}(X_{(N_m-i-1/2)h})\right) - C^i \\
& - \mathcal{E}(h)^{N_m-i-1/2}\left(\nabla f(y_{N_m+i}) - \nabla f(X_{(N_m+i+1/2)h})\right) \\
& - \mathcal{E}(h)^{N_m+i+1/2}\left(\nabla f(y_{N_m-i-1}) - \nabla f(X_{(N_m-i-1/2)h})\right)\Bigg],
\end{aligned}$$

where

$$C_i^j := \mathcal{E}(h)^{N_m+j+1/2}\nabla f_i(X_{(N_m-j-1/2)h}) - 2\mathcal{E}(h)^{N_m}\nabla f_i(X_{N_m h}) + \mathcal{E}(h)^{N_m-j-1/2}\nabla f_i(X_{(N_m+j+1/2)h}),$$

and

$$C^j := \mathcal{E}(h)^{N_m+j+1/2}\nabla f(X_{(N_m-j-1/2)h}) - 2\mathcal{E}(h)^{N_m}\nabla f(X_{N_m h}) + \mathcal{E}(h)^{N_m-j-1/2}\nabla f(X_{(N_m+j+1/2)h}),$$

for $i, j \in \{1, ..., N_m\}$.

Considering the terms excluding the C_i^j and C^i, they can be bounded in L^2 by

$$
\begin{aligned}
&hM\sum_{i=0}^{N_m-1}\Big[\|\tilde{y}_{N_m+i}-X_{(N_m+i+1/2)h}\|_{L^2}+\|\tilde{y}_{N_m-i-1}-X_{(N_m-i-1/2)h}\|_{L^2}\\
&+\|y_{N_m+i}-X_{(N_m+i+1/2)h}\|_{L^2}+\|y_{N_m-i-1}-X_{(N_m-i-1/2)h}\|_{L^2}\Big]\\
&\leq 2hM\sum_{i=0}^{N_m-1}\Big[\Big(\|\tilde{\Delta}_x^{N_m+i}\|_{L^2}+h\|\tilde{\Delta}_v^{N_m+i}\|_{L^2}+h^2\sqrt{Md}\Big)\\
&+\Big(\|\Delta_x^{N_m-i-1}\|_{L^2}+h\|\Delta_v^{N_m-i-1}\|_{L^2}+h^2\sqrt{Md}\Big)\Big].
\end{aligned}
$$

By Lemma 11 (noting that in the full gradient case we can consider Lemma 11 $C_{SG}=0$ and $N_m=1$) we have that the terms, excluding C_i^j and C^i, can be bounded in L^2 by

$$
\begin{aligned}
&hM\sum_{i=0}^{N_m-1}\Big[\|\tilde{y}_{N_m+i}-X_{(N_m+i+1/2)h}\|_{L^2}+\|\tilde{y}_{N_m-i-1}-X_{(N_m-i-1/2)h}\|_{L^2}\\
&+\|y_{N_m+i}-X_{(N_m+i+1/2)h}\|_{L^2}+\|y_{N_m-i-1}-X_{(N_m-i-1/2)h}\|_{L^2}\Big]\\
&\leq \mathcal{O}\Bigg(\sqrt{2}hMN_m\Big[\|\Delta^0\|_{L^2,a,b}+\|\tilde{\Delta}^0\|_{L^2,a,b}\\
&+e^{8hN_m\gamma}\left(h^2N_mM(C_{SG}+1)\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2}+\frac{1}{m}\right)}+\frac{\gamma^2}{M}(\mathbf{C}+h^3N_m\gamma\sqrt{M})\right)\Bigg).
\end{aligned}
$$

We can then use the Itô-Taylor expansion applied to $\mathcal{G}(t)=\mathcal{E}(2N_mh-t)\nabla f_i(X_t)$ to bound the C_i^j as follows

$$
\begin{aligned}
\mathcal{G}(N_mh)=\mathcal{G}((N_m-j-1/2)h)+(j+1/2)h\mathcal{G}'((N_m-j-1/2)h)\\
+\int_0^{(j+1/2)h}\int_0^s d(\mathcal{G}'(s'+(N_m-j-1/2)h))ds,
\end{aligned}
$$

and applying Itô's formula as in Sanz-Serna and Zygalakis (2021) we have

$$
\int_0^{(j+1/2)h}\int_0^s d(\mathcal{G}'(s'+(N_m-j-1/2)h))ds=I_1^{i,j}(h)+I_2^{i,j}(h)+I_3^{i,j}(h)+I_4^{i,j}(h)+I_5^{i,j}(h)
$$

where

$$
\begin{aligned}
I_1^{i,j}(h)&:=\gamma^2\int_0^{(j+1/2)h}\int_0^s\mathcal{E}(2N_m-(s'+(N_m-j-1/2)h))\nabla f_i(X_{s'+(N_m-j-1/2)h})ds'ds\\
I_2^{i,j}(h)&:=\gamma\int_0^{(j+1/2)h}\int_0^s\mathcal{E}(2N_m-(s'+(N_m-j-1/2)h))\nabla^2 f_i(X_{s'+(N_m-j-1/2)h})V_{s'+(N_m-j-1/2)h}ds'ds\\
I_3^{i,j}(h)&:=\\
&\int_0^{(j+1/2)h}\int_0^s\mathcal{E}(2N_m-(s'+(N_m-j-1/2)h))\nabla^3 f_i(X_{s'+(N_m-j-1/2)h})[V_{s'+(N_m-j-1/2)h},V_{s'+(N_m-j-1/2)h}]ds'ds\\
I_4^{i,j}(h)&:=-\int_0^{(j+1/2)h}\int_0^s\mathcal{E}(2N_m-(s'+(N_m-j-1/2)h))\nabla^2 f_i(X_{s'+(N_m-j-1/2)h})\nabla f(X_{s'+(N_m-j-1/2)h})ds'ds\\
I_5^{i,j}(h)&:=\sqrt{2\gamma}\int_0^{(j+1/2)h}\int_0^s\mathcal{E}(2N_m-(s'+(N_m-j-1/2)h))\nabla^2 f_i(X_{s'+(N_m-j-1/2)h})dB_{s'}ds.
\end{aligned}
$$

Then

$$\mathcal{G}(N_m - j - 1/2)h) - 2\mathcal{G}(N_m h) + \mathcal{G}((N_m + j + 1/2)h) = \sum_{k=1}^{5} \left[I_k^{i,j}(2h) - 2I_k^{i,j}(h) \right],$$

and using Lemma 9 we have

$$\begin{aligned}
\|I_1^{i,j}(h)\|_{L^2} &\leq \mathcal{O}\left(h^2 N_m \gamma^2 M \sqrt{\frac{d}{m}}\right), \\
\|I_2^{i,j}(h)\|_{L^2} &\leq \mathcal{O}\left(h^2 N_m \gamma M \sqrt{d}\right), \\
\|I_3^{i,j}(h)\|_{L^2} &\leq \mathcal{O}\left(h^2 N_m M_1 d\right), \\
\|I_4^{i,j}(h)\|_{L^2} &\leq \mathcal{O}\left(h^2 N_m M \sqrt{Md}\right), \\
\|I_5^{i,j}(h)\|_{L^2} &\leq \mathcal{O}\left(h^{3/2} M \sqrt{\gamma N_m d}\right),
\end{aligned}$$

where we remove precise constants for readability. We apply the same bounds for C^i with I_j^i for $j = 1, 2, 3, 4, 5$. We then need to be careful with the $I_5^{i,j}(h)$ terms which are lower order in h individually, but we can use the fact that they have zero expectation conditional on past events to improve the global error bound as in Sanz-Serna and Zygalakis (2021).

Combining all the previous estimates we have that

$$\begin{aligned}
\delta_v^{2N_m} &= \mathcal{E}^{2N_m}(h)\delta_v^0 + \\
&\left[(\star) + h \sum_{i=0}^{N_m - 1} \left(N_m I_5^{N_m - i, i}(2h) - 2N_m I_5^{N_m - i, i}(h) - I_5^i(2h) + 2I_5^i(h) \right) \right] \\
&- h \sum_{i=0}^{N_m - 1} \left(N_m I_5^{N_m - i, i}(2h) - 2N_m I_5^{N_m - i, i}(h) - I_5^i(2h) + 2I_5^i(h) \right).
\end{aligned}$$

Then after $2lN_m$ steps, where $l \in \mathbb{N}$, and is initialized at the target measure we have that

$$\begin{aligned}
\|\delta_v^{2lN_m}\|_{L^2} \leq \mathcal{O}\Bigg(& hMN_m \sum_{k=0}^{l-1} \Bigg[\|\Delta^{2kN_m}\|_{L^2,a,b} + \|\tilde{\Delta}^{2kN_m}\|_{L^2,a,b} \\
& + h^2 N_m M (C_{SG} + 1) e^{8hN_m\gamma} \sqrt{N_m d \left(\frac{\gamma^4 h^2}{m^2} + \frac{1}{m} \right)} \Bigg] \\
& + lh^3 N_m^3 \sqrt{d} \left(\gamma^2 \frac{M}{\sqrt{m}} + M_1 \sqrt{d} \right) + h^{5/2} N_m^{5/2} M \sqrt{l\gamma d} \Bigg).
\end{aligned}$$

where we have used that $\mathbb{E}[I_5^{ij_1} I_5^{ij_2}] = 0$ for $j_1 \neq j_2$.

Position component

Now considering position we write

$$\tilde{x}_{2N_m} - X_{2N_m h} = \underbrace{\tilde{x}_{2N_m} - x_{2N_m}}_{\text{(I)}} + \underbrace{x_{2N_m} - X_{2N_m h}}_{\text{(II)}},$$

and we can bound the (II) using the full gradient scheme bounds in Proposition 8. To make computations easier we consider $\delta_x + \gamma^{-1}\delta_v$ with the previous δ_v bounds. Then we consider (I), the distance to full gradient discretization and we have

$$\begin{aligned}
\delta_x^{2N_m} + \gamma^{-1}\delta_v^{2N_m} &= \delta_x^{2N_m - 1} + \gamma^{-1}\delta_v^{2N_m - 1} - h(N_m \nabla f_1(\tilde{y}_{2N_m - 1}) - \nabla f(y_{2N_m - 1})) \\
&= \delta_x^0 + \gamma^{-1}\delta_v^0 - h\gamma^{-1} \Bigg[\sum_{i=0}^{N_m - 1} (N_m \nabla f_{N_m - i}(\tilde{y}_{N_m + i}) - \nabla f(y_{N_m + i})) \\
&\quad + \sum_{i=0}^{N_m - 1} (N_m \nabla f_{N_m - i}(\tilde{y}_{N_m - i - 1}) - \nabla f(y_{N_m - i - 1})) \Bigg]
\end{aligned}$$

we can bound this similarly to δ_v with the key step being using an Itô-Taylor expansion, but for $\mathcal{G}(t) = \nabla f_i(X_t)$ and we have for $l \in \mathbb{N}$, $\|\delta_x^{2lN_m} + \gamma^{-1}\delta_v^{2lN_m}\|_{L^2}$ can be upper bounded by

$$\mathcal{O}\Bigg(\frac{hMN_m}{\gamma}\sum_{k=0}^{l-1}\Bigg[\|\tilde{\Delta}^k\|_{L^2,a,b} + e^{8hN_m\gamma}\left(h^2N_mM(C_{SG}+1)\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2}+\frac{1}{m}\right)} + \frac{\gamma^2}{M}(\mathbf{C}+h^3N_m\gamma\sqrt{M})\right)\Bigg] + \gamma^{-1}lh^3N_m^3\sqrt{d}\left(\gamma^2\frac{M}{\sqrt{m}}+M_1\sqrt{d}\right) + \gamma^{-1}h^{5/2}N_m^{5/2}M\sqrt{l\gamma d}\Bigg).$$

Therefore

$$\begin{aligned}\|\delta_x^{2lN_m}\|_{L^2} &\leq \|\delta_x^{2lN_m} + \gamma^{-1}\delta_v^{2lN_m}\|_{L^2} + \gamma^{-1}\|\delta_v^{2lN_m}\|_{L^2}\\ &\leq \mathcal{O}\Bigg(\frac{hMN_m}{\gamma}\sum_{k=0}^{l-1}\Bigg[\left(\|\tilde{\Delta}^{2kN_m}\|_{L^2,a,b} + \|\Delta^{2kN_m}\|_{L^2,a,b}\right) + e^{8hN_m\gamma}\left(h^2N_mM(C_{SG}+1)\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2}+\frac{1}{m}\right)} + \frac{\gamma^2}{M}(\mathbf{C}+h^3N_m\gamma\sqrt{M})\right)\Bigg]\\ &\quad + \gamma^{-1}lh^3N_m^3\sqrt{d}\left(\gamma^2\frac{M}{\sqrt{m}}+M_1\sqrt{d}\right) + \gamma^{-1}h^{5/2}N_m^{5/2}M\sqrt{l\gamma d}\Bigg).\end{aligned}$$

Combining components

We now assume that $h < 1/2\gamma N_m$ Therefore combining terms we have that for $l \in \mathbb{N}$ and using the equivalence of norms relation (11) and Proposition 8 we have

$$\begin{aligned}&\|(\tilde{\Delta}_x^{2lN_m}, \tilde{\Delta}_v^{2lN_m})\|_{L^2,a,b} + \|(\Delta_x^{2lN_m}, \Delta_v^{2lN_m})\|_{L^2,a,b} \leq \|(\delta_x^{2lN_m}, \delta_v^{2lN_m})\|_{L^2,a,b} + 2\|(\Delta_x^{2lN_m}, \Delta_v^{2lN_m})\|_{L^2,a,b}\\ &\leq 2\|\delta_x^{2lN_m}|_{L^2} + \frac{2}{\sqrt{M}}\|\delta_v^{2lN_m}\|_{L^2} + \mathcal{O}\left(\frac{\gamma^2}{M}e^{6\max\{\gamma,\frac{2M}{\gamma}\}hlN_m}\left(\mathbf{C}+h^3lN_m\gamma M^{1/2}d^{1/2}\right)\right)\\ &\leq \|\delta_x^{2lN_m} + \gamma^{-1}\delta_v^{2lN_m}\|_{L^2} + \gamma^{-1}\|\delta_v^{2lN_m}\|_{L^2}\\ &\leq \mathcal{O}\Bigg(h\sqrt{M}N_m\sum_{k=0}^{l-1}\left(\|\tilde{\Delta}^{2kN_m}\|_{L^2,a,b} + \|\Delta^{2kN_m}\|_{L^2,a,b}\right)\\ &\quad + lh^3N_m^2M^{3/2}(C_{SG}+1)e^{8hN_m\gamma}\sqrt{N_md\left(\frac{\gamma^4h^2}{m^2}+\frac{1}{m}\right)}\\ &\quad + \frac{lh^3N_m^3\sqrt{d}}{\sqrt{M}}\left(\gamma^2\frac{M}{\sqrt{m}}+M_1\sqrt{d}\right) + h^{5/2}N_m^{5/2}\sqrt{M}\sqrt{l\gamma d}\\ &\quad + \frac{\gamma^2}{M}e^{8hlN_m\sqrt{M}}\left(\mathbf{C}+h^3lN_m\gamma M^{1/2}d^{1/2}\right)\Bigg).\end{aligned}$$

Similarly, if we relax the assumption to $h < 1/2\gamma$ we have the required result. □

Lemma 13. *Assuming that $h < \frac{1}{2\gamma}$ and f is M-∇Lipschitz and is of the form $f = \sum_{i=1}^{N_m} f_i$, $\gamma \geq \sqrt{8M}$ and each f_i is m_i-strongly convex for $i = 1, ..., N_m$. Then considering the SMS-UBU scheme with iterates $(x_i, v_i)_{i\in\mathbb{N}}$ defined by Algorithm 2, approximating the continuous dynamics $(X_t, V_t) \in \mathbb{R}^{2d}$ both initialized from the target measure and having synchronously coupled Brownian motion. Then we have if f is M_1-Hessian Lipschitz that*

$$\|(x_{2lN_m} - X_{2lN_mh}, v_{2lN_m} - V_{2lN_mh})\|_{L^2,a,b} \leq C(\gamma, m, M, M_1, N_m, C_{SG})h^2d,$$

and if f is M_1^s-strongly Hessian Lipschitz

$$\|(x_{2lN_m} - X_{2lN_mh}, v_{2lN_m} - V_{2lN_mh})\|_{L^2,a,b} \leq C(\gamma, m, M, M_1^s, N_m, C_{SG})h^2\sqrt{d}.$$

If we impose the stronger stepsize restriction $h < 1/(12\gamma N_m)$, then our bounds simplify to the form

$$\|(x_{2lN_m} - X_{2lN_m h}, v_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) h^2 N_m^{5/2} \sqrt{d},$$

where $\tilde{\gamma} = \gamma/\sqrt{N_m}, \tilde{m} = m/N_m, \tilde{M} = M/N_m$ and $\tilde{M}_1^s = M_1^s/N_m$.

Proof. Inspired by the interpolation argument used in Leimkuhler et al. (2024c) and also used in Schuh and Whalley (2024) we define (x_{2lN_m}, v_{2lN_m}) as $2lN_m$ steps of the SMS-UBU scheme and (X_{2lhN_m}, V_{2lhN_m}) is defined by (1) at time $2lhN_m \geq 0$, where these are both initialized at $(X_0, V_0) = (x_0, v_0) \sim \pi$ and have synchronously coupled Brownian motion. We further define a sequence of interpolating variants $(\mathbf{X}_{2lN_m}^{(k)}, \mathbf{V}_{2lN_m}^{(k)})$ for every $k = 0, ..., l$ all initialized $(\mathbf{X}_0^{(k)}, \mathbf{V}_0^{(k)}) = (\mathbf{X}_0, \mathbf{V}_0)$, where we define $(\mathbf{X}_{2iN_m}^{(k)}, \mathbf{V}_{2iN_m}^{(k)})_{i=1}^k := (X_{2ihN_m}, V_{2ihN_m})_{i=1}^k$ and $(\mathbf{X}_{2iN_m}^{(k)}, \mathbf{V}_{2iN_m}^{(k)})_{i=k+1}^l$ by SMS-UBU steps (each step being a full forward backward sweep) and for $k = l$ we have, simply, the continuous diffusion (1). Using Proposition 12 we split up the steps into blocks of size $\tilde{l}$ as

$$\begin{aligned}
&\|(x_{2lN_m} - X_{2lN_m h}, v_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq \|(\mathbf{X}_{2lN_m}^{(\lfloor l/\tilde{l} \rfloor \tilde{l})} - \mathbf{X}_{2lN_m h}^{(l)}, \mathbf{V}_{2lN_m}^{(\lfloor l/\tilde{l} \rfloor \tilde{l})} - \mathbf{V}_{2lN_m h}^{(l)})\|_{L^2,a,b} \\
&+ \sum_{j=0}^{\lfloor l/\tilde{l} \rfloor - 1} \|(\mathbf{X}_{2lN_m}^{(j\tilde{l})} - \mathbf{X}_{2lN_m h}^{((j+1)\tilde{l})}, \mathbf{V}_{2lN_m}^{(j\tilde{l})} - \mathbf{V}_{2lN_m h}^{((j+1)\tilde{l})})\|_{L^2,a,b}.
\end{aligned}$$

Next, using the fact that the continuous dynamics preserves the invariant measure, we have contraction of each SMS-UBU forward-backward sweep by Proposition 7 and contraction of each UBU step within each forward-backwards sweep (with different convexity constants). Then we have, by considering each term in the previous summation,

$$\begin{aligned}
&\|(\mathbf{X}_{2lN_m}^{(j\tilde{l})} - \mathbf{X}_{2lN_m h}^{((j+1)\tilde{l})}, \mathbf{V}_{2lN_m}^{(j\tilde{l})} - \mathbf{V}_{2lN_m h}^{((j+1)\tilde{l})})\|_{L^2,a,b} \leq \\
&e^{16h\gamma\tilde{l}N_m} \prod_{i=1}^{N_m} \left(1 - \frac{hm_i N_m}{8\gamma}\right)^{2(l-(j+1)\tilde{l})} C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) \left(lh^3 N_m^4 + h^{5/2}\sqrt{l} N_m^{13/4}\right) \sqrt{d}.
\end{aligned}$$

Summing up the terms we get

$$\begin{aligned}
\|(x_{2lN_m} - X_{2lN_m h}, v_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} &\leq \frac{e^{16h\gamma\tilde{l}N_m} C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) \left(lh^3 N_m^4 + h^{5/2}\sqrt{l} N_m^{13/4}\right) \sqrt{d}}{1 - \prod_{i=1}^{N_m} \left(1 - \frac{hm_i N_m}{8\gamma}\right)^{2\tilde{l}}} \\
&\leq \frac{e^{16h\gamma\tilde{l}N_m} C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) \left(lh^3 N_m^4 + h^{5/2}\sqrt{l} N_m^{13/4}\right) \sqrt{d}}{1 - e^{-\sum_{i=1}^{N_m} \frac{hm_i N_m \tilde{l}}{4\gamma}}} \\
&\leq e^{16h\gamma\tilde{l}N_m} C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG}) \left(lh^3 N_m^4 + h^{5/2}\sqrt{l} N_m^{13/4}\right) \sqrt{d} \left(1 + \frac{4\gamma}{\sum_{i=1}^{N_m} hm_i N_m \tilde{l}}\right),
\end{aligned}$$

If we consider smaller stepsizes, i.e. $h < 1/(2\gamma N_m)$ and choose

$$\tilde{l} = \lceil 1/16hN_m\gamma \rceil$$

and using the fact that $\tilde{l} \leq 1/8hN_m\gamma$ for the considered stepsizes. The right-hand side simplifies to

$$\|(x_{2lN_m} - X_{2lN_m h}, v_{2lN_m} - V_{2lN_m h})\|_{L^2,a,b} \leq C(\gamma/\sqrt{N_m}, m/N_m, M/N_m, M_1^s/N_m) h^2 N_m^{5/2} \sqrt{d},$$

or d rather than $\sqrt{d}$ if the target is Hessian Lipschitz, but not strongly Hessian Lipschitz. These estimates are uniform in l. If we consider the larger stepsize regime $h\gamma N_m \leq 1/2$ we can consider the same argument with $\tilde{l} = \lceil 1/h\gamma \rceil$, but with an exponential dependence on $h\gamma N_m$. □

Theorem 14. *Assuming that $h < \frac{1}{2\gamma}$ and f is M-∇Lipschitz and is of the form $f = \sum_{i=1}^{N_m} f_i$, $\gamma \geq \sqrt{8M}$ and each f_i is m_i-strongly convex for $i = 1, ..., N_m$ with minimizer $x^* \in \mathbb{R}^d$. We consider a SMS-UBU scheme with iterates $(x_k, v_k)_{k\in\mathbb{N}}$ defined by Algorithm 2 with stepsize h, approximating the continuous dynamics $(X_t, V_t) \in \mathbb{R}^{2d}$ both with*

friction parameter γ, initialized at the target measure and having synchronously coupled Brownian motion. We further assume that the stochastic gradients satisfy assumption 2 Then we have if f is M_1-Hessian Lipschitz we have that

$$\|x_k - X_{kh}\|_{L^2} \leq C(\gamma, m, M, M_1, N_m, C_{SG})h^2 d,$$

and if f is M_1^s-strongly Hessian Lipschitz we have that

$$\|x_k - X_{kh}\|_{L^2} \leq C(\gamma, m, M, M_1^s, N_m, C_{SG})h^2\sqrt{d}.$$

If we impose the stronger stepsize restriction $h < 1/(12\gamma N_m)$, then our bounds simplify to the form

$$\|x_k - X_k\|_{L^2} \leq C(\tilde{\gamma}, \tilde{m}, \tilde{M}, \tilde{M}_1^s, C_{SG})h^2 N_m^{5/2}\sqrt{d},$$

where $\tilde{\gamma} = \gamma/\sqrt{N_m}, \tilde{m} = m/N_m, \tilde{M} = M/N_m$ and $\tilde{M}_1^s = M_1^s/N_m$, and similarly when we don't assume the potential is strongly Hessian Lipschitz.

Proof. Using Lemma 9 and defining $(\tilde{\Delta}_x, \tilde{\Delta}_v)$ to be the differences in position and velocity between SMS-UBU and the continuous diffusion, one can show that for $k \in \mathbb{N}$

$$\begin{aligned}\|\tilde{\Delta}_v^k\|_{L^2} &\leq \|\tilde{\Delta}_v^{k-1}\|_{L^2} + hM\|y_{k-1} - X_{(k-1/2)h}\|_{L^2} + h\sqrt{Md} + hM\sqrt{\frac{d}{m}} \\ &\leq (1+h^2M)\|\tilde{\Delta}_v^{k-1}\|_{L^2} + hM\|\tilde{\Delta}_x^{k-1}\|_{L^2} + 3hM\sqrt{\frac{d}{m}},\end{aligned}$$

and similarly

$$\begin{aligned}\|\tilde{\Delta}_x^k\|_{L^2} &\leq \|\tilde{\Delta}_x^{k-1}\|_{L^2} + h\|\tilde{\Delta}_v^{k-1}\|_{L^2} + h^2M\|y_{k-1} - X_{(k-1/2)h}\|_{L^2} + h^2\sqrt{Md} + h^2M\sqrt{\frac{d}{m}} \\ &\leq (1+h^2M)\|\tilde{\Delta}_x^{k-1}\|_{L^2} + h(1+h^2M)\|\tilde{\Delta}_v^{k-1}\|_{L^2} + 3h^2M\sqrt{\frac{d}{m}}.\end{aligned}$$

We then have that

$$\begin{aligned}\|\tilde{\Delta}_x^k\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^k\|_{L^2} &\leq (1+2h\sqrt{M})\left(\|\tilde{\Delta}_x^{k-1}\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^{k-1}\|_{L^2}\right) + 5h\sqrt{\frac{Md}{m}} \\ &\leq e^{2h\sqrt{M}k}\left(\|\tilde{\Delta}_x^0\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^0\|_{L^2}\right) + 5e^{2h\sqrt{M}k}hk\sqrt{\frac{Md}{m}}\end{aligned}$$

and therefore $\|\tilde{\Delta}_v\|_{L^2} \leq \sqrt{M}\left[e^{2h\sqrt{M}k}\left(\|\tilde{\Delta}_x^0\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^0\|_{L^2}\right) + 5e^{2h\sqrt{M}k}hk\sqrt{\frac{Md}{m}}\right]$.

Therefore

$$\begin{aligned}\|\tilde{\Delta}_x^k\|_{L^2} &\leq (1+h^2M)\|\tilde{\Delta}_x^{k-1}\|_{L^2} + h(1+h^2M)\sqrt{M}\left[e^{2h\sqrt{M}k}\left(\|\tilde{\Delta}_x^0\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^0\|_{L^2}\right)\right] \\ &+ 8h^2ke^{2h\sqrt{M}k}(1+h^2M)M\sqrt{\frac{d}{m}} \\ &\leq e^{h^2Mk}\|\tilde{\Delta}_x^0\|_{L^2} + 2e^{3kh\sqrt{M}}\left[hk\sqrt{M}\left(\|\tilde{\Delta}_x^0\|_{L^2} + \frac{1}{\sqrt{M}}\|\tilde{\Delta}_v^0\|_{L^2}\right) + 8(hk)^2M\sqrt{\frac{d}{m}}\right] \\ &\leq \sqrt{2}e^{h^2Mk}\|(\tilde{\Delta}_x^0, \tilde{\Delta}_v^0)\|_{L^2,a,b} + 4e^{3kh\sqrt{M}}\left[hk\sqrt{M}\|(\tilde{\Delta}_x^0, \tilde{\Delta}_v^0)\|_{L^2,a,b} + 8(hk)^2M\sqrt{\frac{d}{m}}\right].\end{aligned}$$

Combining these estimates with Lemma 13 (when considering the iterates of SMS-UBU $(x_i, v_i)_{i\in\mathbb{N}}$ approximating (1) $(X_t, V_t)_{t\geq 0}$) we have for any $i \in \mathbb{N}$ that we can apply the above estimate for $k = i - \lceil \frac{i}{2N_D} \rceil \leq 2N_D$ and then apply Lemma 13 for the remainder and we have the required result. □

Proof of Theorem 5. For $k \in \mathbb{N}$ define $(\hat{x}_k, \hat{v}_k)$ to be SMS-UBU and $(X_t, V_t)_{t\geq 0}$ to be (1), both initialized at the target measure $(\hat{x}_0, \hat{v}_0) := (X_0, V_0) \sim \overline{\pi}$ then,

$$\begin{aligned}
&\|x_k - X_{kh}\|_{L^2} \leq \|x_k - \hat{x}_k\|_{L^2} + \|\hat{x}_k - X_{kh}\|_{L^2} \\
&\leq \sqrt{2}\|(x_k - \hat{x}_k, v_k - \hat{v}_k)\|_{L^2,a,b} + \|\hat{x}_k - X_{kh}\|_{L^2} \\
&\leq \sqrt{2}\exp\left(-\frac{mh}{8\gamma}\left\lfloor\frac{k}{N_m}\right\rfloor\right)\|(x_0 - X_0, v_0 - V_0)\|_{L^2,a,b} \\
&+ \|\hat{x}_k - X_{kh}\|_{L^2},
\end{aligned}$$

where the final term can be bounded by Theorem 14 under appropriate assumptions. □

Theorem 15. *Considering the SG-UBU scheme with friction parameter $\gamma > 0$, stepsize $h > 0$, Markov kernel P and initial measure π_0 and assuming that $h < \frac{1}{2\gamma}$, $\gamma \geq \sqrt{8M}$ and the stochastic gradient satisfies Assumptions 1 and 2 with constants C_G and C_{SG} respectively. Let the potential f be M-∇Lipschitz, m-strongly convex and of the form $f = \sum_{i=1}^{N_m} f_i$. Then at the k-th iteration we have the non-asymptotic bound*

$$\begin{aligned}
\mathcal{W}_{2,a,b}(\pi_0 P^k, \overline{\pi}) \leq &\left(1 - \frac{mh}{4\gamma} + \frac{5h^2 C_G}{M}\right)^{k/2} \mathcal{W}_{2,a,b}(\pi_0, \overline{\pi}) \\
&+ C(\gamma/\sqrt{N_m}, m/N_m, M/N_m, C_G/N_m^2)\left[\frac{C_{SG}\sqrt{h}}{N_m^{1/4}} + h\right]\sqrt{d}.
\end{aligned}$$

Proof. We estimate the difference between the full-gradient UBU scheme and the stochastic gradient scheme UBU scheme initialized at the invariant measure as follows. We use the notation Δ_x^k and Δ_v^k to be the difference in position and velocity at iteration $k \in \mathbb{N}$ respectively. We also use synchronously coupled Brownian motion.

Using (16) and (19) we have

$$\begin{aligned}
\Delta_v^k &= -h\mathcal{E}(h/2)\sum_{i=1}^{k}\mathcal{E}(h)^{k-(i-1)}\left(\nabla f(y_{i-1}) - \mathcal{G}(\tilde{y}_{i-1}, \omega_i)\right), \\
\Delta_x^k &= \mathcal{F}(h)\sum_{i=1}^{k-1}\Delta_v^i - h\mathcal{F}(h/2)\sum_{i=1}^{k}\left(\nabla f(y_{i-1}) - \mathcal{G}(\tilde{y}_{i-1}, \omega_i)\right).
\end{aligned}$$

Then using independence of the stochastic gradient at each iteration we have, from the fact that $\mathcal{G}$ is M-Lipschitz,

$$\|(\Delta_x^k, \Delta_v^k)\|_{L^2,a,b} \leq 4h\sqrt{M}\sum_{i=1}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,b} + \frac{2h}{\sqrt{M}}\sqrt{\sum_{i=1}^{k}\|\nabla f(y_{i-1}) - \mathcal{G}(y_{i-1}, \omega_i)\|^2},$$

and using Assumption 2 we have

$$\leq 4h\sqrt{M}\sum_{i=1}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,b} + \frac{2h}{\sqrt{M}}\sqrt{\sum_{i=1}^{k}C_{SG}^2 M^2\|y_{i-1} - x^*\|^2}.$$

Then using a triangle inequality, the fact that $\|x - x^*\|_{L^2} \leq \sqrt{d/m}$ for $x \sim \pi$, and $\|y_i - X_{(i+1/2)}\|_{L^2} \leq C\frac{M}{m}h\sqrt{d}$ by (Chada et al., 2023, Proposition H.3) we have

$$\begin{aligned}
&\leq 4h\sqrt{M}\sum_{i=1}^{k-1}\|(\Delta_x^i, \Delta_v^i)\|_{L^2,a,b} + \frac{Ch}{\sqrt{M}}\sqrt{kC_{SG}^2 M^2\left(\frac{h^2M^2d}{m^2} + \frac{d}{m}\right)} \\
&\leq \frac{Ch}{\sqrt{M}}e^{4hk\sqrt{M}}\sqrt{kC_{SG}^2 M^2\left(\frac{h^2M^2d}{m^2} + \frac{d}{m}\right)}.
\end{aligned}$$

Using interpolation with blocks of size $\left\lfloor \frac{1}{4h\sqrt{M}} \right\rfloor$ as we did in Lemma 13 with the Wasserstein convergence result of (Whalley, 2024, Proposition B.3.10) we have

$$\begin{aligned}\|(\Delta_x^k, \Delta_v^k)\|_{L^2,a,b} &\leq C\frac{\gamma\sqrt{M}}{m - h\gamma C_G/M}\frac{\sqrt{h}}{\sqrt{M}}\sqrt{\frac{C_{SG}^2 M^2}{\sqrt{M}}\left(\frac{h^2M^2d}{m^2} + \frac{d}{m}\right)} \\ &\leq C(\gamma/\sqrt{N_m}, m/N_m, M/N_m, C_G/N_m^2)\frac{C_{SG}\sqrt{hd}}{N_m^{1/4}},\end{aligned}$$

but we have only bounded the difference between the two schemes and not the diffusion. To conclude the bias estimate we use a triangle inequality with the global error bound for the full-gradient scheme which is a consequence of (Chada et al., 2023, Proposition H.3). We also use the contraction results of (Whalley, 2024, Proposition B.3.10) in combination with the preceeding bias bounds to achieve the non-asymptotic guarantees.

□

D Further details on numerical experiments

D.1 Evaluation of bias

We have evaluated the bias of numerical integrators by synchronously coupling the integrator at step sizes h and $h/2$. For example, if the SG-HMC chain in Algorithm 4 at stepsize $h/2$ uses Gaussians ξ_{2k-1} and ξ_{2k} at steps $2k-1$ and $2k$, then the coupled chain at stepsize h uses $(\xi_{2k-1} + \xi_{2k})/\sqrt{2}$ at step k. Such couplings allowed us to estimate biases more accurately as the variance of the differences becomes significantly smaller compared to using independent chains, while the expected value of the differences remains the same. The maximum stepsize below which stability issues occurred was $h_0 = 2 \cdot 10^{-3}$. Let $h_l = h_0 \cdot 2^{-l}$ for $l \geq 0$. We have constructed couplings between chains with stepsizes $(h_0, h_1), (h_1, h_2), \ldots (h_5, h_6)$, run coupled chains for $400 \cdot 2^l$ epochs for levels (h_l, h_{l+1}), discarded 20% of the samples as burn-in, and used the rest for estimating the difference in expectations between stepsizes h_l and h_{l+1}. We have estimated the full bias using the telescoping sum $\pi_{h_0}(g) - \pi_{h_6}(g) = (\pi_{h_0}(g) - \pi_{h_1}(g)) + \ldots + (\pi_{h_5}(g) - \pi_{h_6}(g))$, assuming that the bias π_{h_6} is negligible. We have estimated the standard deviation of these bias estimates by breaking the total simulation run into 4 chunks of equal size, computing bias estimates based on each one separately, and then computing the standard deviations.

D.2 Behaviour of the norm of Hessian at different points in the weight space

Figure 7 shows the behaviour of the norm of the Hessian for our CNN-based neural network architecture for Fashion-MNIST at different points in the weight space. The norm can be efficiently computed using the power iteration method (Mises and Pollaczek-Geiringer (1929)) based on Hessian-vector products.

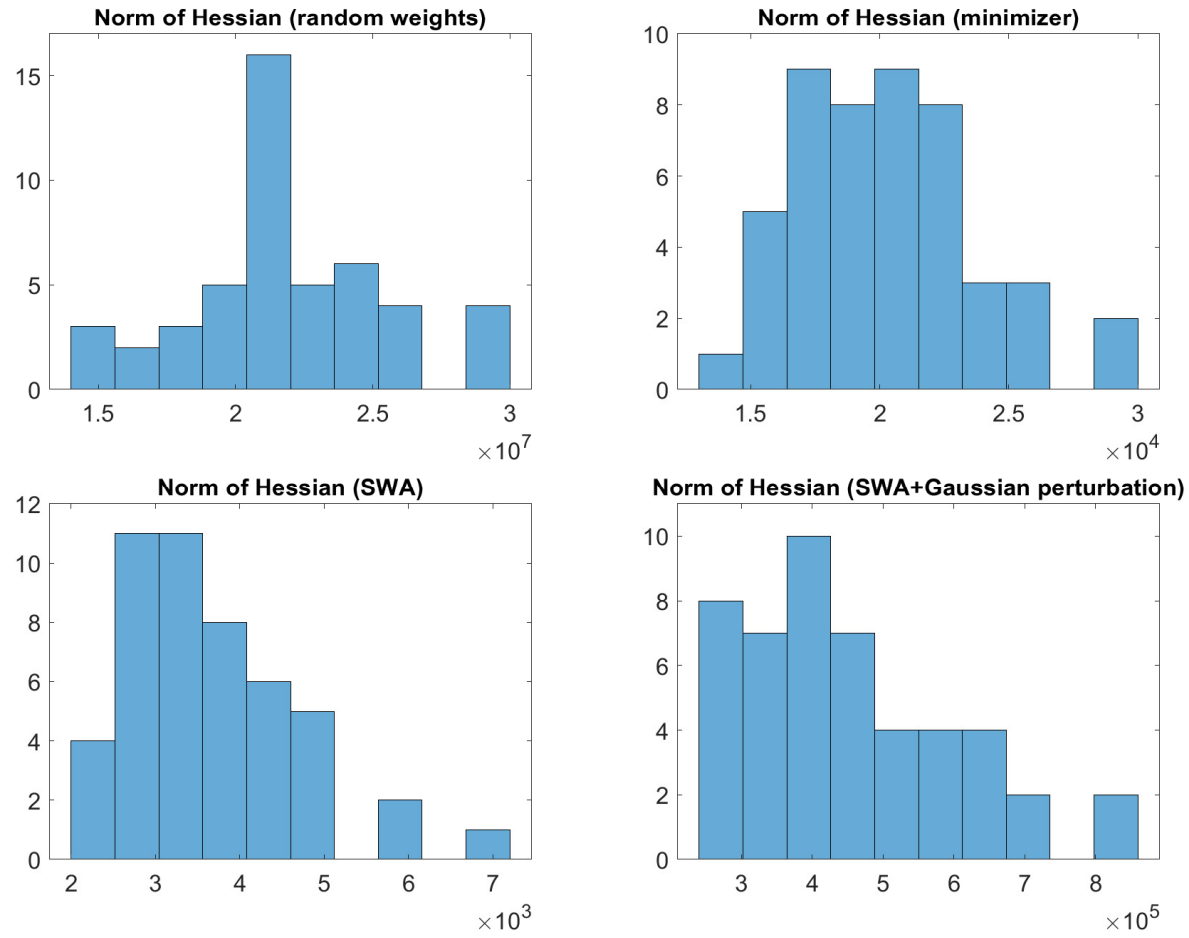

Figure 7: Norm of the Hessian for a CNN-based network for Fashion-MNIST at different points in the weight space over 48 independent runs: random initialisation, minimizer after 15 epochs of training (ADAM), Stochastic Weight Averaging over 5 epochs, and Gaussian perturbation of the SWA network

As we can see, the Hessian has a norm of up to $3 \cdot 10^7$ for a network with random weights, but it has much smaller norm near local minimizers, and especially at the SWA points. Adding some Gaussian noise to the SWA network does lead to a significant increase in the norm of the Hessian, but it is still small compared to using random weights. This observation, together with our theoretical understanding of convergence properties of kinetic Langevin dynamics integrators, leads us to focus on exploring the relatively smooth area in the neighbourhood of the SWA network optima.

E Additional experiments for a wide neural network for CIFAR-10

It has been argued in the literature (Abe et al. (2022)) that the improvements due to the use deep ensembles can be replicated by using wider, larger models with more parameters. We have implemented a wide and large neural network for the CIFAR-10 dataset (Krizhevsky et al. (2009)) (10 classes, 50000 train images, 10000 test images, resolution 32×32, 3 channels). This network is also CNN-based, but it had a number of channels varying from 512 to 1024, which is 4-16 times wider than the networks we used in the other examples. In addition to increasing the width, we have also increased the number of convolutional layers from 6 to 12. The total number of parameters of this network was over 73 million. Pytorch code for this network is provided in Section E.1.

We did not use any data augmentation in this example. Quadratic regularization with precision 1 was used. We have trained this network for 22 epochs, followed by 8 epochs of stochastic weight averaging (SWA) Izmailov et al. (2018), and 60 epochs of SMS-UBU (5 epochs of burn-in, 55 epochs sampling). Our hyperparameter choices are stated in Table 3.

Due to the higher computational cost associated with a larger network, we only performed 4 independent runs with this

Table 3: Choices of hyperparameters for CIFAR-10 experiment

Batch size	Initial L.R.	SWA L.R.	Stepsize h	ρ	$\rho_{\max}$	Friction γ
200	10^{-2}	10^{-3}	$5 \cdot 10^{-4}$	$50^{-1/2}$	$6 \cdot 50^{-1/2}$	ρ^{-1}

network. We did not measure ensemble performance, but only the performance of a single model. Our results are stated in Table 4. As we can see, our BNN implementation based on SMS-UBU improves slightly on the accuracy over SWA and standard training, and it significantly improves in calibration performance (ACE, NLL, and RPS). The accuracy result of 0.8979 obtained using only 90 epochs of training is better than the 0.8964 accuracy obtained using a different network trained with HMC over 60 million epochs in Izmailov et al. (2021).

Table 4: Accuracy and calibration performance of wide neural network model for CIFAR-10, with standard deviations

Training Method	Accuracy (test set)	ACE (test set)	NLL (test set)	RPS (test set)
ADAM	$0.8734(\pm 0.0045)$	$0.0150(\pm 0.0011)$	$0.5475(\pm 0.0352)$	$0.0387(\pm 0.0016)$
SWA	$0.8943(\pm 0.0046)$	$0.0138(\pm 0.0007)$	$0.5620(\pm 0.0507)$	$0.0340(\pm 0.0016)$
BNN (SMS-UBU)	$0.8979(\pm 0.0020)$	$0.0054(\pm 0.0003)$	$0.3086(\pm 0.0061)$	$0.0285(\pm 0.0006)$

E.1 Pytorch code for networks

Pytorch code for neural network of Fashion-MNIST example:

```
in_channels: int = 3
num_classes: int = 2
flattened_size: int = 16384
low_rank: int = 32

self.conv_layer = nn.Sequential(
    # Conv Layer block 1
    nn.Conv2d(in_channels=in_channels, out_channels=32, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(32,momentum=1.0),
    nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
    nn.BatchNorm2d(64,momentum=1.0),
    # Conv Layer block 2
    nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(64,momentum=1.0),
    nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(128,momentum=1.0),
    nn.Conv2d(in_channels=128, out_channels=128, kernel_size=3, padding=1),
    nn.MaxPool2d(kernel_size=2, stride=2),
    # Conv Layer block 3
    nn.BatchNorm2d(128,momentum=1.0),
    nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(256,momentum=1.0),
    nn.Conv2d(in_channels=256, out_channels=256, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
)
```

```
self.fc_layer = nn.Sequential(
    nn.Flatten()
    nn.BatchNorm1d(flattened_size,momentum=1.0),
    nn.Linear(flattened_size, low_rank),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank,512),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(512,momentum=1.0),
)

self.last_layer=nn.Sequential(
    nn.Linear(512, num_classes)
)
```

Pytorch code for neural network of hair colour (blonde/brown) classification on Celeb-A dataset:

```
in_channels: int = 3
num_classes: int = 2
flattened_size: int = 16384
low_rank: int = 16

self.conv_layer = nn.Sequential(
    # Conv Layer block 1
    nn.Conv2d(in_channels=in_channels, out_channels=32, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(32,momentum=1.0),
    nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
    nn.BatchNorm2d(64,momentum=1.0),
    # Conv Layer block 2
    nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(128,momentum=1.0),
    nn.Conv2d(in_channels=128, out_channels=128, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
    nn.BatchNorm2d(128,momentum=1.0),
    # Conv Layer block 3
    nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(256,momentum=1.0),
    nn.Conv2d(in_channels=256, out_channels=256, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
)

self.fc_layer = nn.Sequential(

    nn.BatchNorm1d(flattened_size,momentum=1.0),
    nn.Linear(flattened_size, low_rank),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank, 4096),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(4096,momentum=1.0),
    nn.Linear(4096, low_rank),
```

```
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank, 1024),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(1024,momentum=1.0),
    nn.Linear(1024, low_rank),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank, 512),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(512,momentum=1.0),
)

self.last_layer=nn.Sequential(
    nn.Linear(512, num_classes)
)
```

Pytorch code for neural network of Chest X-ray example:

```
in_channels: int = 1
num_classes: int = 2
flattened_size: int = 12544
low_rank: int = 32

self.conv_layer = nn.Sequential(
    nn.Conv2d(in_channels=in_channels, out_channels=32, kernel_size=3, stride=1, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=1),
# Second Convolutional Block
    nn.BatchNorm2d(32,momentum=1.0),
    nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, stride=1, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=1),
# Third Convolutional Block
    nn.BatchNorm2d(64,momentum=1.0),
    nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, stride=1, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=1),
# Fourth Convolutional Block
    nn.BatchNorm2d(64,momentum=1.0),
    nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=1),
# Fifth Convolutional Block
    nn.BatchNorm2d(128,momentum=1.0),
    nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, stride=1, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=1),
)

self.fc_layer = nn.Sequential(
    nn.Flatten()
    nn.BatchNorm1d(flattened_size,momentum=1.0),
    nn.Linear(flattened_size, low_rank),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank, 512),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(512,momentum=1.0),
```

```
)

self.last_layer=nn.Sequential(
    nn.Linear(512, num_classes)
)
```

Pytorch code for wide neural network of CIFAR-10 example (73038346 parameters):

```
in_channels: int = 3
num_classes: int = 10
flattened_size: int = 16384
low_rank: int = 512

self.conv_layer = nn.Sequential(
nn.Sequential(
    nn.Conv2d(in_channels=in_channels, out_channels=512, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(512,momentum=1.0),
    nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(512,momentum=1.0),
    nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(512,momentum=1.0),
    nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
    nn.BatchNorm2d(512,momentum=1.0),


    # Conv Layer block 2
    nn.Conv2d(in_channels=512, out_channels=768, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(768,momentum=1.0),
    nn.Conv2d(in_channels=768, out_channels=768, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(768,momentum=1.0),
    nn.Conv2d(in_channels=768, out_channels=768, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(768,momentum=1.0),
    nn.Conv2d(in_channels=768, out_channels=768, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
    nn.BatchNorm2d(768,momentum=1.0),

    #Conv Layer block 3
    nn.Conv2d(in_channels=768, out_channels=1024, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(1024,momentum=1.0),
    nn.Conv2d(in_channels=1024, out_channels=1024, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(1024,momentum=1.0),
    nn.Conv2d(in_channels=1024, out_channels=1024, kernel_size=3, padding=1),
    nn.Softplus(beta=1.0),
    nn.BatchNorm2d(1024,momentum=1.0),
    nn.Conv2d(in_channels=1024, out_channels=1024, kernel_size=3, padding=1),
```

Dataset	Training size	Test size	Resolution	Channels	# of param.
Fashion-MNIST	60000	10000	28×28	1	1265258
Celeb-A	55400	6073	64×64	3	1612306
Chest X-ray	5217	600	180×180	1	870882

Table 5: Details about the datasets and neural networks used in our experiments

```
    nn.Softplus(beta=1.0),
    nn.MaxPool2d(kernel_size=2, stride=2),
)

self.fc_layer = nn.Sequential(
    nn.Flatten()
    nn.BatchNorm1d(flattened_size,momentum=1.0),
    nn.Linear(flattened_size, low_rank),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank,2048),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(2048,momentum=1.0),
    nn.Linear(2048,low_rank),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(low_rank,momentum=1.0),
    nn.Linear(low_rank,1024),
    nn.Softplus(beta=1.0),
    nn.BatchNorm1d(1024),
)

self.last_layer=nn.Sequential(
    nn.Linear(1024, num_classes)
)
```

F Comparison with Symmetric Minibatch Splitting HMC

In Cobb and Jalaian (2021), a symmetric minibatch splitting version of HMC was proposed, including a Metropolis-Hastings accept/reject step. As a comparison, we have implemented Metropolised method this for the same multinomial logistic regression example of Section 3.4, using the same variance reduction scheme on the potentials, using the same batch size and dataset. We have also tried a slightly modified version of the scheme of Cobb and Jalaian (2021), stated in Algorithm 7. This version is different from the original scheme in terms of the type of leapfrog steps used (first a half step in position update, followed by a full step in velocity update, and then another half step in position update), and the partial velocity refreshment step (as in Generalized HMC (Horowitz (1991))). At the same step sizes, we found that Algorithm 7 had significantly higher acceptance rates compared to the original scheme in Cobb and Jalaian (2021). We did some tuning of the parameters, and obtained the best performance $h = 10^{-5}$, $L = 10$, and $\alpha = 0.7$. With $K = 1000$ iterations, the acceptance rate was 0.844. We discarded 20% of the samples as burn-in, and computed the calibration performance based on the remaining ones. Accuracy was 0.8420, ACE was 0.0195, NLL was 0.4464, and RPS was 0.0391.

Comparing with Figure 1 on page 7 of the submission, we can see that the calibration results are essentially the same as for SMS-UBU and the other unadjusted methods, confirming that both have very low bias. Nevertheless, the step size $h = 10^{-5}$ ($\log_2 h \approx -16.61$) for SMS-GHMC is 100 times smaller than the step size $h = 10^{-3}$ at which SMS-UBU was already offering essentially the same calibration performance. Hence the Metropolized method SMS-GHMC is much more computationally expensive in this example compared to the unadjusted ones based on second-order integrators.

F.1 Further details for calibration experiments

The dataset sizes for our experiments are stated below. In the Fashion-MNIST and Celeb-A examples, we did not use any data augmentation. For the chest X-ray experiments, we have used random rotations, shifts, cropping, and brightness

Algorithm 7 Symmetric Minibatch Splitting Generalized HMC (SMS-GHMC)

Initialize $(x_0, v_0) \in \mathbb{R}^{2d}$, stepsize $h > 0$, number of minibatches N_m, number of sweeps per accept/reject step L, partial refreshment parameter $\alpha \in [0, 1)$, number of iterations K.
for $k = 1, 2, ..., K$ **do**
  Sample $\omega_1, ..., \omega_{N_m} \in [N_D]^{N_b}$ uniformly without replacement
  $(x, v) \to (x_{k-1}, v_{k-1})$
  for $l = 1, 2, \dots, L$ **do**
    Forward Sweep
    for $b = 1, 2, ..., N_m$ **do**
      $x \to x + \frac{h}{2} v$
      $v \to v - h\mathcal{G}(x, \omega_b)$
      $x \to x + \frac{h}{2} v$
    end for
    Backward Sweep
    for $b = 1, 2, ..., N_m$ **do**
      $x \to x + \frac{h}{2} v$
      $v \to v - h\mathcal{G}(x, \omega_{N_m+1-b})$
      $x \to x + \frac{h}{2} v$
    end for
  end for
  Metropolis-Hastings Accept/Reject Step
  Compute difference of Hamiltonians $H(x_{k-1}, v_{k-1}) - H(x, v) = f(x_{k-1}) + \frac{\|v_{k-1}\|^2}{2} - f(x) - \frac{\|v\|^2}{2}$.
  Sample $U_k \sim \text{Uniform}[0, 1]$.
  if $\log(U_k) < H(x_{k-1}, v_{k-1}) - H(x, v)$ **then**
    $(x_k, v_k) = (x, v)$
  else
    $(x_k, v_k) = (x_{k-1}, -v_{k-1})$.
  end if
  Velocity Refreshment Step
  Sample $Z_k \sim \mathcal{N}(0_d, I_d)$
  Update $v_k \to \alpha v_k + \sqrt{1 - \alpha^2} Z_k$
end for
Output: Samples $(x_k)_{k=0}^K$.

changes to augment the dataset size to 47400, and then used those images for training.

Batch size was chosen as 200 in each case. In all datasets, quadratic regularisation with variance 1 was applied on all weights except biases. An important implementation detail is that the batch normalization layers were not using running means, but only the current batch. This was done to ensure a well-defined loss function. To ensure good performance in the prediction task, we have evaluated the network a single larger batch consisting of 20 individual batches (4000 images) before predicting with network weights set to evaluation mode. In the Bayesian setting, no temperature parameter was used, but the potential was simply chosen as the total cross-entropy loss plus the regulariser.

Our hyperparameter choices are stated in Table 6.

Table 6: Details about the datasets and neural networks used in our experiments

Dataset	Initial L.R.	SWA L.R.	Stepsize h	ρ	$\rho_{\max}$	Friction γ
Fashion-MNIST	10^{-2}	10^{-3}	$2.5 \cdot 10^{-4}$	$50^{-1/2}$	$6 \cdot 50^{-1/2}$	ρ^{-1}
Celeb-A(blonde/brun.)	10^{-2}	10^{-3}	$2.5 \cdot 10^{-4}$	0.2	$6 \cdot 0.2$	ρ^{-1}
Chest X-ray	10^{-2}	10^{-3}	$5 \cdot 10^{-4}$	$50^{-1/2}$	$6 \cdot 50^{-1/2}$	ρ^{-1}