DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS

doi:10.3934/dcdss.2021157

IMPROVING SAMPLING ACCURACY OF STOCHASTIC GRADIENT MCMC METHODS VIA NON-UNIFORM SUBSAMPLING OF GRADIENTS

Ruilin Li

Georgia Institute of Technology, USA

Xin Wang

Google Research, USA

Hongyuan Zha

School of Data Science, Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong, Shenzhen, China

Molei Tao*

Georgia Institute of Technology, USA

Abstract. Many Markov Chain Monte Carlo (MCMC) methods leverage gradient information of the potential function of target distribution to explore sample space efficiently. However, computing gradients can often be computationally expensive for large scale applications, such as those in contemporary machine learning. Stochastic Gradient (SG-)MCMC methods approximate gradients by stochastic ones, commonly via uniformly subsampled data points, and achieve improved computational efficiency, however at the price of introducing sampling error. We propose a non-uniform subsampling scheme to improve the sampling accuracy. The proposed exponentially weighted stochastic gradient (EWSG) is designed so that a non-uniform-SG-MCMC method mimics the statistical behavior of a batch-gradient-MCMC method, and hence the inaccuracy due to SG approximation is reduced. EWSG differs from classical variance reduction (VR) techniques as it focuses on the entire distribution instead of just the variance; nevertheless, its reduced local variance is also proved. EWSG can also be viewed as an extension of the importance sampling idea, successful for stochastic-gradient-based optimizations, to sampling tasks. In our practical implementation of EWSG, the non-uniform subsampling is performed efficiently via a Metropolis-Hastings chain on the data index, which is coupled to the MCMC algorithm. Numerical experiments are provided, not only to demonstrate EWSG's effectiveness, but also to guide hyperparameter choices, and validate our *non-asymptotic global error bound* despite of approximations in the implementation. Notably, while statistical accuracy is improved, convergence speed can be comparable to the uniform version, which renders EWSG a practical alternative to VR (but EWSG and VR can be combined too).

2020 *Mathematics Subject Classification.* Primary: 65C05, 65C40, 60J20; Secondary: 62D05, 37A50, 37A60, 90C15.

Key words and phrases. MCMC sampling and Bayesian inference, non-uniform weighted stochastic gradient, transition kernel matching, nonasymptotic sampling error bound, bias-variance trade-off.

*Corresponding author.

© 2021 The Author(s). Published by AIMS, LLC. This is an Open Access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. **Introduction.** Consider the construction of algorithms that sample a target probability distribution $\pi \sim Z^{-1}\rho(\boldsymbol{x})d\boldsymbol{x}$, where Z is a normalization constant and the unnormalized density ρ is assumed to be nonzero on the domain. Let $V(\boldsymbol{x}) := -\log \rho(\boldsymbol{x})$ and then the target density can be rewritten in the form of Gibbs distribution, i.e. $Z^{-1}\exp(-V(\boldsymbol{x}))$, where V will be referred to as the potential function.

For this purpose, many MCMC algorithms use physics-inspired evolution such as Langevin dynamics [9] to utilize gradient information (i.e., ∇V) in order to efficiently explore the target distribution over continuous parameter space. However, gradient-based MCMC methods are often limited by the computational cost of evaluating the gradient on large data sets, which often correspond to specific potentials of the form $V(\boldsymbol{x}) = \sum_{i=1}^{n} V_i(\boldsymbol{x})$, where n is very large; this type of additive potential with many terms will be the setup of this paper.

Motivated by the great success of stochastic gradient methods for optimization, which uses a stochastic estimator of the batch gradient ∇V instead of evaluating all ∇V_i terms, stochastic gradient MCMC methods (SG-MCMC) for sampling have also been gaining increasing attention. More precisely, when the accurate but expensive-to-evaluate batch gradients in a MCMC method are replaced by computationally cheaper estimates based on a subset of the data, the method is turned to a stochastic gradient version. Classical examples include SG (overdamped) Langevin Dynamics [42] and SG Hamiltonian Monte Carlo [12], both designed for scalability suitable for machine learning tasks.

However, directly replacing the batch gradient by a (uniform) stochastic one without additional mitigation generally causes a MCMC method to sample from a statistical distribution different from the target, because the transition kernel of the MCMC method gets corrupted by the noise of subsampled gradient. In general, the additional noise is tolerable if the learning rate/step size is tiny or decreasing. However, when large steps are used for better efficiency, the extra noise is non-negligible and undermines the performance of downstream applications such as Bayesian inference.

In this paper, we present a state-dependent non-uniform SG-MCMC algorithm termed **E**xponentially **W**eighted **S**tochastic **G**radients method (EWSG), which continues the efforts of uniform SG-MCMC methods for scalable sampling. Our approach is based on designing the transition kernel of a SG-MCMC method to approximate the transition kernel of a full-gradient-based MCMC method. This approximation leads to non-uniform (in fact, exponential) weights that aim at capturing the entire state-variable distribution of the full-gradient-based MCMC method, rather than providing unbiased gradient estimator and reducing its variance. Nevertheless, if focusing on the variance, the advantage of EWSG is the following: recall the stochasticity of a SG-MCMC method can be decomposed into the *intrinsic* randomness of MCMC and the *extrinsic* randomness introduced by gradient subsampling; in conventional uniform subsampling treatments, the latter randomness is independent of the former, and thus when they are coupled together, variances add up; EWSG, on the other hand, dynamically chooses the weight of each datum according to the current state of the MCMC, and thus the variances do not add up due to dependence. However, the gained accuracy is beyond reduced variance, as EWSG, when converged, samples from a distribution close to the invariant distribution of the full-gradient MCMC method (which has no variance contributed by the extrinsic randomness), because its transition kernel (of the corresponding

Markov process) is close to that of the full-gradient-MCMC method. This is how better sampling accuracy can be achieved.

Our main demonstration of EWSG is based on 2nd-order Langevin equations (a.k.a. inertial, kinetic, or underdamped Langevin), although it works for other MCMC methods too (e.g., Appendix E,F). To concentrate on the role of non-uniform SG weights, we will work with constant step sizes only. The fact that EWSG has locally reduced variance than its uniform counterpart is rigorously shown in Theorem 4.2. Furthermore, a global non-asymptotic error analysis is given in Theorem 4.3 to quantify the convergence and improved accuracy of EWSG, as well as to provide insights about hyperparameter choices.

Practically, the non-uniform gradient subsampling of EWSG is efficiently implemented via a Metropolis-Hastings chain over the data index. A number of experiments on synthetic and real world data sets, across downstream tasks including Bayesian logistic regression and Bayesian neural networks, are conducted to demonstrate the effectiveness of EWSG and validate our theoretical results, despite the approximation used in the implementation. In addition to improved accuracy, the convergence speed was empirically observed, in a fair comparison setup based on the same data pass, to be comparable to its uniform counterpart when hyperparameters are appropriately chosen. The convergence (per data pass) was also seen to be clearly faster than a classical Variance Reduction (VR) approach (note: for sampling, not optimization), and EWSG hence provides a useful alternative to VR. Additional theoretical study of EWSG convergence speed is provided in Appendix H.

Notation-wise, ∇V will be referred to as the full/batch-gradient, $n\nabla V_I$ with random $I \in [n]$, which is a statistical estimator of ∇V, will be called stochastic gradient (SG), and when I is uniformly distributed it will be called a uniform SG/subsampling, otherwise non-uniform. When uniform SG is used to approximate the batch-gradient in underdamped Langevin, the method will be referred to as (vanilla) Stochastic Gradient Underdamped Langevin Dynamics (SGULD/SGHMC[1]), and it serves as a baseline in experiments.

2. Related works.

Stochastic Gradient MCMC Methods (SG-MCMC). Based on approximating gradients by uniformly subsampled ones, stochastic gradient methods are computationally more favorable than their full gradient counterparts and have been widely studied and used in the field of optimization. Inspired by the great success of stochastic gradient methods in optimization, people also have also applied stochastic gradient methods to sampling problems. Since the seminal work of Stochastic Gradient Langevin Dynamics (SGLD) [42], much progress [1, 34] has been made in the field of SG-MCMC. [40] theoretically justified the convergence of SGLD and offered practical guidance on tuning step size. [25] introduced a preconditioner and improved stability of SGLD. We also refer to [30] and [20] which will be discussed in Sec. 5. While these work were mostly based on 1st-order (overdamped) Langevin, other dynamics were considered too. For instance, [12] proposed Stochastic Gradient Hamiltonian Monte Carlo (SGHMC), which is closely related to 2nd-order Langevin dynamics [8, 6], and [29] put it in a more general framework. 2nd-order

[1]SGULD is the same as the well-known SGHMC with $\hat{B} = 0$, see eq. (13) and Sec. 3.3 in [12] for details. To be consistent with existing literature, we will refer SGULD as SGHMC in the sequel.

Langevin was recently shown to be faster than the 1st-order version in appropriate setups [14, 13, 27] and began to gain more attention.

Variance Reduction (VR). For optimization, vanilla SG methods usually find approximate solutions quickly but the convergence slows down (due to variance) when an accurate solution is needed [2, 22]. SAG [38] improved the convergence speed of stochastic gradient methods to linear, which is the same as gradient descent methods with full gradient, at the expense of large memory overhead. SVRG [22] successfully reduced this memory overhead. SAGA [17] furthers improved convergence speed over SAG and SVRG. For sampling, [19] applied VR techniques to SGLD (see also [3, 10]). However, many VR methods have large memory overhead and/or periodically use the whole data set for gradient estimation calibration, and hence can be resource-demanding.

EWSG is derived based on matching transition kernels of MCMC and improves the accuracy of the entire distribution rather than just the variance. However, it does have a consequence of variance reduction and thus can be implicitly regarded as a VR method. When compared to the classic work on VR for SG-MCMC [19], EWSG converges faster when the same amount of data pass is used, although its sampling accuracy is below that of VR for Gaussian targets (but well above vanilla SG; see Sec. 5.1). In this sense, EWSG and VR suit different application domains: EWSG can replace vanilla SG for tasks in which the priority is speed and then accuracy, as it keeps the speed but improves the accuracy; on the other hand, VR remains to be the heavy weapon for accuracy-demanding scenarios. Importantly, EWSG, as a generic way to improve SG-MCMC methods, can be combined with VR too (e.g., Sec. F); thus, they are not exclusive or competing with each other.

Importance Sampling (IS). IS methods employ nonuniform weights to improve the convergence speed of stochastic gradient methods for optimization. Traditional IS methods use fixed weights that do not change along iterations, and the weight computation requires prior information of gradient terms, e.g., Lipschitz constant of the gradient [33, 37, 15], which are usually unknown or difficult to estimate. Adaptive IS was also proposed in which the importance was re-evaluated at each iteration, whose computation usually required the entire data set per iteration and may also require information like the upper bound of gradient [46, 47].

For sampling, it is not easy to combine IS with SG [20]; the same paper is, to our knowledge, the closest to this goal and will be compared with in Sec. 5.3. EWSG can be viewed as a way to combine (adaptive) IS with SG for efficient sampling. It require no oracle about the gradient, nor any evaluation over the full data set. Instead, an inner-loop Metropolis chain maintains a random index that approximates a state-dependent non-uniform distribution (i.e. the weights/importance).

Other Mini-batch MCMC Methods. Besides SG-MCMC methods, there are also many non-gradient-based MCMC methods that use only a subset of data in each iteration so that the MCMC methods can scale to large data sets. For example, austerity MH [23] formulates Metropolis-Hastings step as a statistical hypothesis testing problem and proposes to use only a subset of data to make statistically significant accept/reject decision. Using a subsampled unbiased estimator of the likelihood in a pseudo-marginal framework to accelerate the Metropolis-Hastings algorithm is proposed in [4]. A notable *exact* MCMC method is FlyMC [30], which introduces an auxiliary binary random variable for each datum and only the subset

of data whose corresponding auxiliary binary indicator "light" up, are used in iteration. Some more recent advances on exact MCMC methods include [45, 44]. We also refer to [4] for an excellent review on subsampling MCMC methods.

3. **Underdamped Langevin: The continuous time backbone of a MCMC method.** Underdamped Langevin Dynamics (ULD) is given by the SDE

$$\begin{cases} d\boldsymbol{\theta} &= \boldsymbol{r}dt \\ d\boldsymbol{r} &= -(\nabla V(\boldsymbol{\theta}) + \gamma \boldsymbol{r})dt + \sigma d\boldsymbol{W} \end{cases} \tag{1}$$

where $\boldsymbol{\theta}, \boldsymbol{r} \in \mathbb{R}^d$ are state and momentum variables, V is a potential energy function which in our context is, as originated from cost minimization or Bayesian inference over many data, the sum of many terms $V(\boldsymbol{\theta}) = \sum_{i=1}^n V_i(\boldsymbol{\theta})$, γ is a friction coefficient, σ is intrinsic noise amplitude, and $\boldsymbol{W}$ is a standard d-dimensional Wiener process. Under mild assumptions on V, Langevin dynamics admits a unique invariant distribution $\pi(\boldsymbol{\theta}, \boldsymbol{r}) \sim \exp\left(-\frac{1}{T}(V(\boldsymbol{\theta}) + \frac{\|\boldsymbol{r}\|^2}{2})\right)$ and is in many cases geometric ergodic [35]. T is the temperature of system determined via the fluctuation dissipation theorem $\sigma^2 = 2\gamma T$ [24].

We consider ULD instead of the overdamped version mainly for two reasons: (i) one may think ULD is more complicated, and we'd like to show it is still easy to be paired with EWSG (EWSG can work for many MCMC methods; Appendix E has an overdamped version); (ii) it is believed that ULD has faster convergence than overdamped Langevin for instance in high-dimensions where (local) condition number is likely to be larger (e.g., [14, 13, 39]). Like the overdamped version, numerical integrators for ULD with well captured statistical properties of the continuous process have been extensively investigated (e.g, [36, 7]), and both the overdamped and underdamped integrators are friendly to derivations that will allow us to obtain explicit expressions of the non-uniform weights.

4. **Method.**

4.1. **Motivation: An illustration of non-optimality of uniform subsampling.** Uniform subsampling of gradients have long been the dominant way of stochastic gradient approximations mainly because it is intuitive, unbiased and easy to implement.

However, uniform gradient subsampling can introduce large noise, and is suboptimal even in the family of unbiased stochastic gradient estimator, as the following Theorem 4.1 will show. One intuition is, consider for example cases where data size n is larger than dimension d. In such cases, $\{\nabla V_i\}_{i=1,2,\cdots,n} \subset \mathbb{R}^d$ are linearly dependent and hence it is likely that there exist probability distributions $\{p_i\}_{i=1,2,\cdots,n}$ other than the uniform one such that the gradient estimate is unbiased, however with smaller variance because linearly dependent terms need not to be all used. This is a motivation for us to develop non-uniform subsampling schemes (weights may be $\boldsymbol{\theta}$ dependent), although we will not require $n > d$ later.

Theorem 4.1. *Suppose given $\boldsymbol{\theta} \in \mathbb{R}^d$, the errors of SG approximation $\boldsymbol{b}_i = n\nabla V_i(\boldsymbol{\theta}) - \nabla V(\boldsymbol{\theta}), 1 \le i \le n$ are i.i.d. absolutely continuous random vectors with possibly-$\boldsymbol{\theta}$-dependent density $p(\cdot|\boldsymbol{\theta})$ and $n > d$. We call $\boldsymbol{p} \in \mathbb{R}^n$ a sparse vector if the number of non-zero entries in $\boldsymbol{p}$ is no greater than $d+1$, i.e. $\|\theta\|_0 \le d+1$. Then with probability 1, the optimal probability distribution $\boldsymbol{p}^\star$ that is unbiased and*

minimizes the trace of the covariance of $n\nabla V_I(\boldsymbol{\theta})$*, i.e.* $\boldsymbol{p}^\star$ *which solves the following, is a sparse vector.*

$$\min_{\boldsymbol{p}} \mathrm{Tr}(\mathbb{E}_{I\sim\boldsymbol{p}}[\boldsymbol{b}_I\boldsymbol{b}_I^T]) \quad s.t.\ \mathbb{E}_{I\sim\boldsymbol{p}}[\boldsymbol{b}_I] = \boldsymbol{0}, \tag{2}$$

Despite the sparsity of $\boldsymbol{p}^\star$, which seemingly suggests one only needs at most $d+1$ gradient terms per iteration when using SG methods, it is not practical because $\boldsymbol{p}^\star$ requires solving the linear programming problem (2) in Theorem 4.1, for which an entire data pass is needed. Nevertheless, this result motivates us to seek alternatives to uniform SG. For example, the EWSG method we will develop will have reduced local variance with high probability, and at the same time remain efficiently implementable without having to use all data per parameter update; it can be biased though, but a global error analysis (Thm.4.3) will show that trading bias for variance can still be worthy.

4.2. **Exponentially weighted stochastic gradient.** MCMC methods are characterized by their transition kernels. In traditional SG-MCMC methods, uniform SG is used, which is independent of the intrinsic randomness of MCMC methods (e.g. diffusion in ULD), as a result, the transition kernel of SG-MCMC is quite different from that with full gradient. Therefore, it is natural to ask — is it possible to couple the two originally independent randomness, so that the transition kernel of the SG-MCMC better matches that of the batch-gradient-MCMC, and the sampling accuracy is thus improved?

Here is one way to do so. Consider Euler-Maruyama (EM) discretization[2] of Eq. (1):

$$\begin{cases} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \boldsymbol{r}_k h \\ \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - (\nabla V(\boldsymbol{\theta}_k) + \gamma \boldsymbol{r}_k)h + \sigma\sqrt{h}\boldsymbol{\xi}_{k+1} \end{cases} \tag{3}$$

where h is step size and $\boldsymbol{\xi}_{k+1}$'s are i.i.d. d-dimensional standard Gaussian random variables. Denote the transition kernel of EM discretization with full gradient by $P^{EM}(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k, \boldsymbol{r}_k)$.

Then consider a SG version: replace $\nabla V(\boldsymbol{\theta}_k)$ by a weighted SG $n\nabla V_{I_k}(\boldsymbol{\theta}_k)$, where I_k is the index chosen to approximate full gradient and has p.m.f. $\mathbb{P}(I_k = i|\boldsymbol{\theta}_k, \boldsymbol{r}_k) = p_i$. Denote the new transition kernel by $\tilde{P}^{EM}(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k, \boldsymbol{r}_k)$.

It is not hard to see that

$$\begin{aligned} &P^{EM}(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k, \boldsymbol{r}_k) \\ =&1_{\{\boldsymbol{\theta}_k+\boldsymbol{r}_k h\}}(\boldsymbol{\theta}_{k+1})\,\frac{1}{Z}\exp\left(-\frac{\|\boldsymbol{r}_{k+1} - \boldsymbol{r}_k + (\nabla V(\boldsymbol{\theta}_k) + \gamma\boldsymbol{r}_k)h\|^2}{2\sigma^2 h}\right) \\ =&1_{\{\boldsymbol{\theta}_k+\boldsymbol{r}_k h\}}(\boldsymbol{\theta}_{k+1})\,\frac{1}{Z}\exp\left(-\frac{\|\boldsymbol{x} + \sum_{i=1}^n \boldsymbol{a}_i\|^2}{2}\right) \end{aligned}$$

and

$$\tilde{P}^{EM}(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k, \boldsymbol{r}_k) = 1_{\{\boldsymbol{\theta}_k+\boldsymbol{r}_k h\}}(\boldsymbol{\theta}_{k+1})\,\frac{1}{\tilde{Z}}\sum_{j=1}^n p_i \exp\left(-\frac{\|\boldsymbol{x} + n\boldsymbol{a}_i\|^2}{2}\right),$$

[2]EM is not the most accurate or robust discretization, see e.g., [36, 7], but since it may still be the most used method, demonstrations here will be based on EM. The same idea of EWSG can easily apply to most other discretizations such as GLA [7].

where Z and $\tilde{Z}$ are normalization constants, $\boldsymbol{x} \triangleq \frac{\boldsymbol{r}_{k+1}-\boldsymbol{r}_k+h\gamma\boldsymbol{r}_k}{\sigma\sqrt{h}}$ and $\boldsymbol{a}_i \triangleq \frac{\sqrt{h}\nabla V_i(\boldsymbol{\theta}_k)}{\sigma}$. From these two expressions, one can see that if we could choose

$$p_i \propto \exp\left(-\frac{\|\boldsymbol{x}+\sum_{i=1}^n \boldsymbol{a}_i\|^2}{2}+\frac{\|\boldsymbol{x}+n\boldsymbol{a}_i\|^2}{2}\right), \tag{4}$$

we would have $P^{EM}(\boldsymbol{\theta}_{k+1},\boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{r}_k) = \tilde{P}^{EM}(\boldsymbol{\theta}_{k+1},\boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{r}_k)$ and be able to recover the transition kernel of full gradient with that of stochastic gradient. However, Eq.(4) is only formal and infeasible, because $\boldsymbol{x}$ is dependent on the future state variable $\boldsymbol{r}_{k+1}$ which we do not know. Therefore, to obtain a practically implementable algorithm, we will fix $\boldsymbol{x}$ as a hyper-parameter and hope that the approximation is good enough so that we still have $P^{EM}(\boldsymbol{\theta}_{k+1},\boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{r}_k) \approx \tilde{P}^{EM}(\boldsymbol{\theta}_{k+1},\boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{r}_k)$.

We refer to the choice of p_i in eq.4 Exponentially Weighted Stochastic Gradient (**EWSG**). Unlike Thm.4.1, EWSG does not require $n > d$ to work. Note the idea of designing non-uniform weights of SG-MCMC to match the transition kernel of full gradient can be suitably applied to a wide class of gradient-based MCMC methods; for example, Sec. E shows how EWSG can be applied to Langevin Monte Carlo (overdamped Langevin), and Sec. F shows how it can be combined with VR. Therefore, EWSG complements a wide range of SG-MCMC methods.

Since the weight choice of EWSG is motivated by approximating the transition kernel of a full-gradient MCMC method, we anticipate EWSG to be statistically more accurate than a uniformly-subsampled stochastic gradient estimator. As a special but commonly interested accuracy measure, the smaller variance of EWSG is shown with high probability[3]:

Theorem 4.2. *Assume $\{\nabla V_i(\boldsymbol{\theta})\}_{i=1,2,\cdots,n}$ are i.i.d random vectors and $|\nabla V_i(\boldsymbol{\theta})| \leq R$ for some constant R almost surely. Denote the uniform distribution over $[n]$ by $\boldsymbol{p}^U$, the exponentially weighted distribution by $\boldsymbol{p}^E$, and let $\Delta = \mathrm{Tr}[\mathrm{cov}_{I\sim\boldsymbol{p}^E}[n\nabla V_I(\boldsymbol{\theta})|\boldsymbol{\theta}] - \mathrm{cov}_{I\sim\boldsymbol{p}^U}[n\nabla V_I(\boldsymbol{\theta})|\boldsymbol{\theta}]]$. If $\boldsymbol{x} = \mathcal{O}(\sqrt{h})$, we have $\mathbb{E}[\Delta] < 0$, and $\exists C > 0$ independent of n or h such that $\forall\epsilon > 0$,*

$$\mathbb{P}(|\Delta - \mathbb{E}[\Delta]| \geq \epsilon) \leq 2\exp\left(-\frac{\epsilon^2}{nCh^2}\right).$$

It is not surprising that less non-intrinsic local variance correlates with better global statistical accuracy, which will be made explicit and rigorous in the next subsection.

4.3. **Non-asymptotic error bound.** We now establish a non-asymptotic global sampling error bound (in mean square distance between arbitrary test observables) of SG underdamped Langevin algorithms (the bound applies to both EWSG and other methods e.g., SGHMC). The full proof is deferred to the Appendix C, but the main tool we will be using is the Poisson equation machinery [32, 41, 11]. A brief overview is the following:

[3]'With high probability' but not almost surely because Theorem 4.2 in fact suits a class of weights, which includes but is not limited to EWSG.

Let $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{r} \end{pmatrix}$. The generator $\mathcal{L}$ of diffusion process (1) is

$$\begin{aligned}\mathcal{L}(f(\boldsymbol{X}_t)) = &\lim_{h\to 0} \frac{\mathbb{E}[f(\boldsymbol{X}_{t+h})] - \mathbb{E}[f(\boldsymbol{X}_t)]}{h} \\ =&\boldsymbol{r}^T \nabla_{\boldsymbol{\theta}} f - (\gamma \boldsymbol{r} + \nabla V(\boldsymbol{\theta}))^T \nabla_{\boldsymbol{r}} f + \gamma \Delta_{\boldsymbol{r}} f.\end{aligned}$$

Given a test function $\phi(\boldsymbol{x})$, its posterior average is $\bar{\phi} = \int \phi(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$, approximated by its time average of samples $\widehat{\phi}_K = \frac{1}{K}\sum_{k=1}^K \phi(\boldsymbol{X}_k^E)$, where $\boldsymbol{X}_k^E$ is the sample path given by EM integrator. Then the Poisson equation $\mathcal{L}\psi = \phi - \bar{\phi}$ can be a useful tool for the weak convergence analysis of SG-MCMC. The solution ψ characterizes the difference between ϕ and its posterior average $\bar{\phi}$.

Our main theoretical result is the following:

Theorem 4.3. *Assume* $\mathbb{E}[\|\nabla V_i(\boldsymbol{\theta}_k^E)\|^l] < M_1, \mathbb{E}[\|\boldsymbol{r}_k^E\|^l] < M_2, \forall l = 1, 2, \cdots, 12, \forall i = 1, 2, \cdots, n$ *and* $\forall k \geq 0$. *Assume the solution to the Poisson equation,* ψ*, exists, and its derivatives up to 3rd-order are uniformly bounded* $\|D^l\psi\|_\infty < M_3, l = 0, 1, 2, 3$. *Then there exist constants* $C_1, C_2, C_3 > 0$ *depending on* M_1, M_2, M_3*, such that*

$$\mathbb{E}\big(\widehat{\phi}_K - \bar{\phi}\big)^2 \leq C_1 \frac{1}{T} + C_2 \frac{h}{T} \frac{\sum_{k=0}^{K-1} \mathbb{E}[\mathrm{Tr}[\mathrm{cov}(n\nabla V_{I_k}|\mathcal{F}_k)]]}{K} + C_3 h^2 \tag{5}$$

where $T = Kh$ *is the corresponding time in the underlying continuous dynamics,* I_k *is the index of the datum used to estimate the gradient at* k*-th iteration, and* $cov(n\nabla V_{I_k}|\mathcal{F}_k)$ *is the covariance of stochastic gradient at* k*-th iteration conditioned on the current sigma algebra* $\mathcal{F}_k$ *in the filtration.*

Remark 1. (*interpreting the three terms in the bound*) Unlike a typical VR method which aims at finding unbiased gradient estimator with reduced variance, EWSG aims at bringing the entire density closer to that of a batch-gradient MCMC. As a consequence, its practical implementation may correspond to SG that has reduced variance but a small bias too. Eq.(5) quantifies this bias-variance trade-off. How the extrinsic local variance and bias contribute to the global error is respectively reflected in the 2nd and 3rd terms, although the 3rd term also contains a contribution from the numerical discretization error. With or without bias, the 3rd term remains $\mathcal{O}(h^2)$ because of this discretization error. However, for moderate T, the 2nd term is generally larger than the 3rd due to its lower order in h, which means reducing local variance can improve sampling accuracy even if at the cost of introducing a small bias. Since EWSG has a smaller local variance than uniform SG (Thm.4.2, as a special case of improved overall statistical accuracy), its global performance is also favorable. The 1st term is for the convergence of the continuous process (eq.1 in this case).

Remark 2. (*innovation and relation with the literature*) Thm.4.3, to the best of our knowledge, is the first that incorporates the effects of both local bias and local variance of a SG approximation (previous SOTA bounds are only for unbiased SG). It still works when restricting to unbiased SG, and in this case our bound reduces to SOTA [41, 11]. Some more facts include: [32], being the seminal work from which we adapt our proof, only discussed the batch gradient case, whereas our theory has additional (non-uniform) SG. [41, 11] studied the effect of SG, but the SG considered there did not use state-dependent weights, which would destroy several martingales used in their proofs. Unlike in [32] but like in [41, 11], our state space is not the compact torus but $\mathbb{R}^d$. Also, the time average $\widehat{\phi}_K$, to which our results

apply, is a commonly used estimator, particularly when using a long time trajectory of Markov chain for sampling. However, if one is interested in an alternative of using an ensemble for sampling, techniques in [14, 16] might be useful to further bound difference between the law of $\boldsymbol{X}_k$ and the target distribution.

4.4. **Practical implementation.** In EWSG, the probability of each gradient term is $p_i = \widehat{Z}^{-1} \exp\left\{-\frac{\|\boldsymbol{x}+\sum_{j=1}^n \boldsymbol{a}_j\|^2}{2} + \frac{\|\boldsymbol{x}+n\boldsymbol{a}_i\|^2}{2}\right\}$. Although the term $\|\boldsymbol{x}+\sum_{j=1}^n \boldsymbol{a}_j\|^2/2$ depends on the full data set, it is shared by all p_i's and can be absorbed into the normalization constant $\hat{Z}^{-1}$ (we still included it explicitly due to the needs in proofs); unique to each p_i is only the term $\|\boldsymbol{x} + n\boldsymbol{a}_i\|^2/2$. This motivates us to run a Metropolis-Hastings chain over the possible indices $i \in \{1, 2 \cdots, n\}$: at each inner-loop step, a proposal of index j is uniformly drawn, and then accepted with probability $P(i \to j) =$

$$\min\left\{1, \exp\left(\frac{\|\boldsymbol{x} + n\boldsymbol{a}_j\|^2}{2} - \frac{\|\boldsymbol{x} + n\boldsymbol{a}_i\|^2}{2}\right)\right\}; \tag{6}$$

if accepted, the current index i is replaced by j. When the chain converges, the index will follow the distribution given by p_i. The advantage is, we avoid passing through the entire data sets to compute each p_i, but the index will still approximately sample from the non-uniform distribution.

In practice, we often perform only $M = 1$ step of the Metropolis index chain per integration step, especially if h is not too large. The rationale is, when h is small, the outer iteration evolves slower than the index chain, and as θ does not change much in, say, N outer steps, effectively $N \times M$ inner steps take place on almost the same index chain, which makes the index r.v. equilibrate better. Regarding the larger h case (where the efficacy of local variance reduction via non-uniform subsampling is more pronounced; see e.g., Thm. 4.3), $M = 1$ may no longer be optimal, but improved sampling with large h and $M = 1$ is still clearly observed in various experiments (Sec. 5).

Another hyper-parameter is $\boldsymbol{x}$ (see earlier discussion in Sec.4.2). Our heuristic recommendation is $\boldsymbol{x} = \frac{\sqrt{h}\gamma \boldsymbol{r}_k}{\sigma}$. The rationale is, as long as $r_{k+1} - r_k$'s density is maximized at 0 (which will be the case at least for large k as r_k will converge to a Gaussian), this choice of $\boldsymbol{x}$ is a maximum likelihood estimator. This approximation appeared to be a good one in all our experiments with medium h and $M = 1$.

Sec. 5.1 further investigates hyperparameter selection empirically and shows that approximations due to M and $\boldsymbol{x}$ is not detrimental to our non-asymptotic theory in Sec. 4.3.

Practical EWSG is summarized in Algorithm 1. For simplicity of notation, we restrict the description to mini batch size $b = 1$, but an extension to $b > 1$ is straightforward. See Sec. D in appendix. Practical EWSG has reduced variance but does not completely eliminate the extrinsic noise created by SG due to its approximations. A small bias was also created by these approximations, but its effect is dominated by the variance effect (see Sec. 4.3). In practice, if needed, one can combine EWSG with other VR technique to further improve accuracy. Appendix F describes how EWSG can be combined with SVRG.

5. **Experiments.** In this section, the proposed EWSG algorithm will be compared with SGHMC [12], SGLD [42], as well as several more recent popular approaches, including FlyMC [30], pSGLD [25], CP-SGHMC [20] (a method closest to the goal of

Algorithm 1 EWSG

Input: {the number of data terms n, gradient functions $V_i(\cdot), i = 1, 2, \cdots, n$, step size h, the number of data passes K, index chain length M, friction and noise coefficients γ and σ}
Initialize $\boldsymbol{\theta}_0, \boldsymbol{r}_0$ (arbitrarily, or use an informed guess)
for $k = 0, 1, \cdots, \lceil \frac{Kn}{M+1} \rceil$ **do**
  $i \leftarrow$ uniformly sampled from $1, \cdots, n$, compute and store $n\nabla V_i(\boldsymbol{\theta}_k)$
  $I \leftarrow i$
  for $m = 1, 2, \cdots, M$ **do**
    $j \leftarrow$ uniformly sampled from $1, \cdots, n$, compute and store $n\nabla V_j(\boldsymbol{\theta}_k)$
    $I \leftarrow j$ with probability in Equation 6
  end for
  Evaluate $\tilde{V}(\boldsymbol{\theta}_k) = nV_I(\boldsymbol{\theta}_k)$
  Update $(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}) \leftarrow (\boldsymbol{\theta}_k, \boldsymbol{r}_k)$ via one step of Euler-Maruyama integration using $\tilde{V}(\boldsymbol{\theta}_k)$
end for

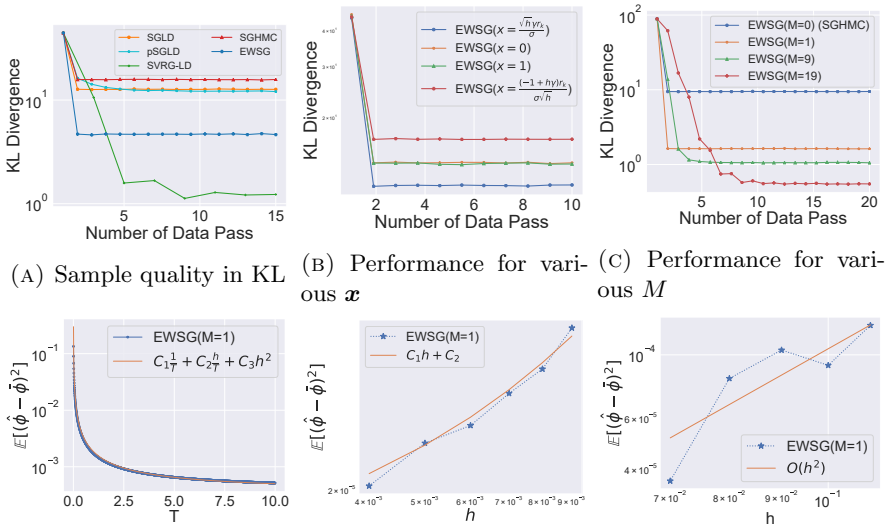
applying IS idea to SG-based sampling) and SVRG-LD [19] (overdamped Langevin improved by VR). Sec. 5.1 is a detailed empirical study of EWSG on simple models, with comparison and implication of two important hyper-parameters M and $\boldsymbol{x}$, and verification of the non-asymptotic theory (Theorem 4.3). Sec. 5.2 demonstrates EWSG for Bayesian logistic regression on a large-scale data set. Sec. 5.3 is a Bayesian Neural Network (BNN) example. It serves only as a high-dimensional, multi-modal test case, and we do not intend to compare Bayesian and non-Bayesian neural nets. As FlyMC requires a tight lower bound of likelihood, known for only a few cases, it will only be compared against in Sec. 5.2 where such a bound is obtainable. CP-SGHMC requires heavy tuning on the number of clusters which differs across data sets/algorithms, so it will only be included in the BNN example, for which the authors empirically found a good hyper parameter for MNIST [20]. SVRG-LD is only compared to in Sec. 5.1, because SG-MCMC methods can converge within only one data pass in Sec. 5.2, rendering control-variate based VR technique inapplicable, and it was suggested that VR leads to poor results for deep models (e.g., Sec. 5.3) [18].

For fair comparison, all algorithms use constant step sizes and are allowed fixed computation budget, i.e., for L data passes, all algorithms can only call gradient function nL times. All experiments are conducted on a machine with a 2.20GHz Intel(R) Xeon(R) E5-2630 v4 CPU and an Nvidia GeForce GTX 1080 GPU. If not otherwise mentioned, $\sigma = \sqrt{2\gamma}$ so only γ needs specification, the length of the index chain is set $M = 1$ for EWSG and the default values of two hyper-parameters required in pSGLD are set $\lambda = 10^{-5}$ and $\alpha = 0.99$, as suggested in [25].

5.1. **Gaussian examples.** Consider sampling from a simple 2D Gaussian whose potential function is

$$V(\boldsymbol{\theta}) = \sum_{i=1}^{n} V_i(\boldsymbol{\theta}) = \sum_{i=1}^{n} \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{c}_i\|^2.$$

We set $n = 50$ and randomize $\boldsymbol{c}_i$ from a two-dimensional standard normal $\mathcal{N}(\mathbf{0}, I_2)$. Due to the simplicity of $V(\boldsymbol{\theta})$, we can write the target density analytically and will



(A) Sample quality in KL

(B) Performance for various $\boldsymbol{x}$

(C) Performance for various M

(D) MSE against time T (1^{st} and 2^{nd} terms in Eq. (5))

(E) MSE against step size h with fixed finite T (2^{nd} term in Eq. (5))

(F) MSE against step size h with $T \approx \infty$ (3^{rd} term in Eq. (5))

FIGURE 1. Performance quantification (Gaussian target)

use KL divergence $\mathrm{KL}(p\|q) = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}$ to measure the difference between the target distribution and generated samples.

For each algorithm, we generate 10,000 independent realizations for empirical estimation. All algorithms are run for 30 data passes with minibatch size of 1. Step size is tuned from $5 \times \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and 5×10^{-3} is chosen for SGLD and pSGLD, 5×10^{-2} for SGHMC and EWSG and 5×10^{-4} for SVRG-LD. SGHMC and EWSG use $\gamma = 10$. Results are shown in Fig. 1a and EWSG outperforms SGHMC, SGLD and pSGLD in terms of accuracy. Note SVRG-LD has the best accuracy[4] but the slowest convergence, and that is why EWSG is a useful alternative to VR: its light-weight suits situations with limited computational resources better.

Figure 1b shows the performance of several possible choices of the hyper-parameter $\boldsymbol{x}$, including the recommended option $\boldsymbol{x} = \sqrt{h}\gamma \boldsymbol{r}_k/\sigma$, and $\boldsymbol{x} = \mathbf{0}$, $\boldsymbol{x} = \mathbf{1}$, $\boldsymbol{x} = (-1 + h\gamma)\boldsymbol{r}_k/\sigma\sqrt{h}$ (which corresponds to $\boldsymbol{r}_{k+1} = \mathbf{0}$). Step size $h = 7 \times 10^{-2}$ is used for this experiment. The recommended option performs better than the others.

Another important hyper-parameter in EWSG is M. As the length of index chain M increases, the subsampling distribution approaches that given by Eq.4. Considering that larger M means more gradient evaluations per step[5], there could be some M value that achieves the best balance between speed and accuracy. Fig.1c shows a fair comparison of four values of $M = 0, 1, 9, 19$, and the recommended $M = 1$ case converges as fast as SGHMC (when $M = 0$, EWSG does not run the

[4]For Gaussians, mean and variance completely determine the distribution, so appropriately reduced variance leads to great accuracy for the entire distribution.

[5]in each iteration of the outer MCMC loop, EWSG consumes $M+1$ data points, and hence in a fair comparison with fixed computation budget (e.g. E total gradient calls), EWSG runs $\frac{E}{M+1}$ iterations which is decreasing in M.

Method	SGLD	pSGLD	SGHMC	EWSG	FlyMC
Accuracy(%)	75.283 ± 0.016	75.126 ± 0.020	75.268 ± 0.017	**75.306 ± 0.016**	75.199 ± 0.080
Log Likelihood	-0.525 ± 0.000	-0.526 ± 0.000	-0.525 ± 0.000	**-0.523 ± 0.000**	**-0.523 ± 0.000**
Wall Time (s)	**3.085 ± 0.283**	4.312 ± 0.359	3.145 ± 0.307	3.755 ± 0.387	291.295 ± 56.368

TABLE 1. Accuracy, log likelihood and wall time of various algorithms on test data after one data pass (mean ± std).

Metropolis-Hastings index chain and hence degenerates to SGHMC) but improves its accuracy. It is also clear that as M increases, sampling accuracy gets improved.

As approximations are used in Algorithm 1, it is natural to ask if results of Thm. 4.3 still hold. We empirically investigate this question (using $M = 1$ and variance as the test function ϕ). Eq.5 in Thm.4.3 is a nonasymptotic error bound consisting of three parts, namely an $\mathcal{O}(\frac{1}{T})$ term corresponding to the convergence at the continuous limit, an $\mathcal{O}(h/T)$ term coming from the SG variance, and an $\mathcal{O}(h^2)$ term due to bias and numerical error. Fig.1d plots the mean squared error (MSE) against time $T = Kh$ to confirm the 1st term. Fig.1e plots the MSE against h with fixed T in the small h regime (so that the 3rd term is negligible when compared to the 2nd) to confirm that the 2nd term scales like $\mathcal{O}(h)$. For the 3rd term in Eq. (5), we run sufficiently many iterations to ensure all chains are well-mixed, and Fig.1f confirms the final MSE to scale like $\mathcal{O}(h^2)$ even for large h (as the 2nd term vanishes due to $T \to \infty$). In this sense, despite the approximations introduced by the practical implementation, the performance of Algorithm 1 is still approximated by Thm. 4.3, even when $M = 1$. Thm. 4.3 can thus guide the choices of h and T in practice.

5.2. **Bayesian logistic regression (BLR).** Consider Bayesian logistic regression for the binary classification problem. The probabilistic model for predicting a label y_k given a feature vector x_k is $p(y_k = 1|\boldsymbol{x}_k, \boldsymbol{\theta}) = 1/(1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x}_k))$. We set a Gaussian prior with zero mean and covariance $\Sigma = 10 I_d$ for $\boldsymbol{\theta}$, and hence the potential function of the posterior distribution of $\boldsymbol{\theta}$ is

$$V(\boldsymbol{\theta}) = 5\|\boldsymbol{\theta}\|^2 - \sum_{i=1}^{n} y_i \log p(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\theta}) + (1 - y_i) \log\left(1 - p(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\theta})\right).$$

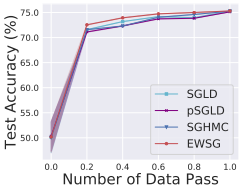
We conduct our experiments on Covertype data set[6], which contains $n = 581,012$ data points and 54 features (which is the dimension of $\boldsymbol{\theta}$). Given the large size of this data set, SG is needed to scale up MCMC methods. We use 80% of data for training and the rest 20% for testing.

The FlyMC algorithm[7] uses a lower bound derived in [30] for likelihood function. For underdamped Langevin based algorithms, we set friction coefficient $\gamma = 50$. After tuning, we set the step size as $\{1, 3, 0.02, 5, 5\} \times 10^{-3}$ for SGULD, EWSG, SGLD, pSGLD and FlyMC. All algorithms are run for one data pass, with minibatch size of 50 (for FlyMC, it means 50 data are sampled in each iteration to switch state). 100 independent samples are drawn from each algorithm to estimate statistics. To further smooth out noise, all experiments are repeated 1000 times with different seeds.

Results are in Fig. 2a and 2b and Table 1. EWSG outperforms others, except for log likelihood being comparable to FlyMC, which is an *exact* MCMC method.

[6] https://archive.ics.uci.edu/ml/datasets/covertype

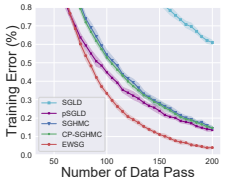
[7] https://github.com/HIPS/firefly-monte-carlo/tree/master/flymc



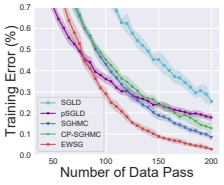
(A) Test Accuracy

(B) Test Log Likelihood

FIGURE 2. BLR learning curve



(A) MLP architecture



(B) CNN architecture

FIGURE 3. BNN learning curve. Shade: one standard deviation.

The wall time consumed by EWSG is only slightly more than that of SGLD and SGHMC, but less than pSGLD and orders-of-magnitude less than FlyMC.

5.3. **Bayesian neural network (BNN).** Bayesian neural network is a compelling model for deep learning [43]. Here two popular architectures of BNN are experimented – multilayer perceptron (MLP) and convolutional neural nets (CNN). In MLP, a hidden layer with 100 neurons followed by a softmax layer is used. In CNN, we use standard network configuration with 2 convolutional layers followed by 2 fully connected layers [21]. Both convolutional layers use 5×5 convolution kernel with 32 and 64 channels, 2×2 max pooling layers follow immediately after convolutional layer. The last two fully-connected layers each has 200 neurons. We set the standard normal as prior for all weights and bias.

We test algorithms on the MNIST data set, consisting of 60,000 training data and 10,000 test data, each datum is a 28×28 gray-scale image with one of the ten possible labels (digits $0 \sim 9$). For ULD based algorithms , we set friction coefficient $\gamma = 0.1$ in MLP and $\gamma = 1.0$ in CNN. In MLP, the step sizes are set $h = \{4, 2, 2\} \times 10^{-3}$ for EWSG, SGHMC and CP-SGHMC, and $h = \{0.001, 1\} \times 10^{-4}$ for SGLD and pSGLD, via grid search. For CP-SGHMC , (clustering-based preprocessing is conducted [20] before SGHMC) we use K-means with 10 clusters to preprocess the data set. In CNN, the step sizes are set $h = \{4, 2, 2\} \times 10^{-3}$ for EWSG, SGHMC and CP-SGHMC, and $h = \{0.02, 8\} \times 10^{-6}$ for SGLD and pSGLD, via grid search. All algorithms use minibatch size of 100 and are run for 200 epoches. For each algorithm, we generate 100 independent samples to make posterior prediction. To smooth out noise and obtain more significant results, we repeat experiments 10 times with different seeds.

The learning curve of training error is shown in Fig. 3a and 3b. EWSG consistently improves over its uniform counterpart (i.e., SGHMC) and CP-SGHMC (an approximate IS SG-MCMC). Moreover, EWSG also outperforms two standard benchmarks SGLD and pSGLD. The improvement over baseline on MNIST data set is comparable to some of the early works [12, 25].

Note: in the MLP setup, the model has $d > 78,400$ parameters whereas there are $n = 60,000$ data points, which shows EWSG does not require $n > d$ to work and can still outperform its uniform counterpart in the overparametrized regime (Thm.4.1 demonstrates the underparametrized case only because the sparsity result is easy to understand, but EWSG doesn't only work for underparameterized models).

Acknowledgments. We thank Yian Ma and Xuefeng Gao for insightful discussions. MT is grateful for the partial support by NSF DMS-1847802 and ECCS-1936776. Part of the research of HZ is supported by Shenzhen Research Institute of Big Data. This work was mainly conducted when XW was a PhD student and HZ was a professor, both at Georgia Institute of Technology.

Appendix A. Proof of Theorem 4.1.

Proof. Denote the set of all n-dimensional probability vectors by Σ^n, the set of sparse probability vectors by $\mathcal{S}$, and the set of non-sparse (dense) probability vectors by $\mathcal{D} = \Sigma^n \setminus \mathcal{S}$. Denote $B = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n]$, then the optimization problem can be written as

$$\min \sum_{i=1}^{n} p_i \|\boldsymbol{b}_i\|^2$$
$$\text{s.t.} \begin{cases} B\boldsymbol{p} = \boldsymbol{0} \\ \boldsymbol{p}^T \boldsymbol{1}_n = 1 \\ p_i \geq 0, i = 1, 2, \cdots, n \end{cases}$$

Note that the feasible region is always non-empty (take $\boldsymbol{p}$ to be a uniform distribution) and is also closed and bounded, hence this linear programming is always solvable. Denote the set of all minimizers by $\mathcal{M}$. Note that $\mathcal{M}$ depends on $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n$ and is in this sense random.

The Lagrange function is

$$L(\boldsymbol{p}, \boldsymbol{\lambda}, \mu, \boldsymbol{\omega}) = \boldsymbol{p}^T \boldsymbol{s} - \boldsymbol{\lambda}^T B\boldsymbol{p} - \mu(\boldsymbol{p}^T \boldsymbol{1}_n) - \boldsymbol{\omega}^T \boldsymbol{p}$$

where $\boldsymbol{s} = [\|\boldsymbol{b}_1\|^2, \|\boldsymbol{b}_2\|^2, \cdots, \|\boldsymbol{b}_n\|^2]^T$ and $\boldsymbol{\lambda}, \mu, \boldsymbol{\omega}$ are dual variables. The optimality condition reads as

$$\frac{\partial L}{\partial \boldsymbol{p}} = \boldsymbol{s} - B^T \boldsymbol{\lambda} - \mu \boldsymbol{1}_n - \boldsymbol{\omega} = \boldsymbol{0}$$

Dual feasibilty and complementary slackness require

$$\omega_i \leq 0, i = 1, 2, \cdots, n$$
$$\boldsymbol{\omega}^T \boldsymbol{p} = 0$$

Consider the probability of the event {a dense probability vector can solve the above minimization problem}, i.e., $\mathbb{P}(\mathcal{M} \cap \mathcal{D} \neq \emptyset)$. It is upper bounded by

$$\mathbb{P}(\mathcal{M} \cap \mathcal{D} \neq \emptyset) \leq \mathbb{P}(\boldsymbol{p} \in \mathcal{D} \text{ and } \boldsymbol{p} \text{ solves KKT condition})$$

Since $\boldsymbol{p} \in \mathcal{D}$, complementary slackness implies that at least $d + 2$ entries in $\boldsymbol{\omega}$ are zero. Denote the indices of these entries by $\mathcal{J}$. For every $j \in \mathcal{J}$, by optimality condition, we have $s_j - \boldsymbol{\lambda}^T \boldsymbol{b}_j - \mu = 0$, i.e.,

$$\|\boldsymbol{b}_j\|^2 - \boldsymbol{\lambda}^T \boldsymbol{b}_j - \mu = 0$$

Take the first $d + 1$ indices in $\mathcal{J}$, and note a geometric fact that $d + 1$ points in a d-dimensional space must be on the surface of a hypersphere of at most $d - 1$ dimension, which we denote by $\mathcal{S} = S^{q-1} + \boldsymbol{x}$ for some vector $\boldsymbol{x}$ and integer $q \leq d$. Because b_i's distribution is absolutely continuous, we have

$$\begin{aligned} &\mathbb{P}(\boldsymbol{p} \in \mathcal{D} \text{ and } \boldsymbol{p} \text{ solves KKT condition}) \\ \leq &\mathbb{P}(\boldsymbol{p} \in \mathcal{D} \text{ and } \boldsymbol{b}_j \in S, \forall j \in \mathcal{J}) \\ \leq &\mathbb{P}(\boldsymbol{b}_j \in S, \forall j \in \mathcal{J}) \end{aligned}$$

$$
\begin{aligned}
&=\mathbb{P}(\boldsymbol{b}_{j_k} \in S, k = d+2, \cdots, |\mathcal{J}|) \\
&= \prod_{k=d+2}^{|\mathcal{J}|} \mathbb{P}(\boldsymbol{b}_{j_k} \in S) \qquad \text{(independence)} \\
&=0 \qquad \text{(absolute continuous)}
\end{aligned}
$$

Hence $\mathbb{P}(\mathcal{M} \cap \mathcal{D} \neq \emptyset) = 0$ and

$$
\begin{aligned}
1 &= \mathbb{P}(\mathcal{M} \neq \emptyset) \\
&= \mathbb{P}((\mathcal{M} \cap \mathcal{S}) \cup (\mathcal{M} \cap \mathcal{D}) \neq \emptyset) \\
&\leq \mathbb{P}(\mathcal{M} \cap \mathcal{S} \neq \emptyset) + \mathbb{P}(\mathcal{M} \cap \mathcal{D} \neq \emptyset) \\
&= \mathbb{P}(\mathcal{M} \cap \mathcal{S} \neq \emptyset)
\end{aligned}
$$

Therefore we have

$$\mathbb{P}(\mathcal{M} \cap \mathcal{S} \neq \emptyset) = 1$$

□

Appendix B. Proof of Theorem 4.2.

Proof. Let $\boldsymbol{b}_i = n\nabla V_i$ and assume $\|\boldsymbol{b}_i\|_2 \leq R$ for some constant R. Denote $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n]$. For any probability distribution $\boldsymbol{p}$ over $\{1, \cdots, n\}$, we have

$$
\begin{aligned}
&\operatorname{cov}_{I\sim\boldsymbol{p}}[\boldsymbol{b}_I|\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n] \\
=&\sum_{i=1}^n p_i \boldsymbol{b}_i \boldsymbol{b}_i^T - \left(\sum_{i=1}^n p_i \boldsymbol{b}_i\right)\left(\sum_{i=1}^n p_i \boldsymbol{b}_i\right)^T \\
=&\sum_{i=1}^n p_i \boldsymbol{b}_i \boldsymbol{b}_i^T \sum_{i=1}^n p_i - \left(\sum_{i=1}^n p_i \boldsymbol{b}_i\right)\left(\sum_{i=1}^n p_i \boldsymbol{b}_i\right)^T \\
=&\sum_{i<j} (\boldsymbol{b}_i - \boldsymbol{b}_j)(\boldsymbol{b}_i - \boldsymbol{b}_j)^T p_i p_j
\end{aligned}
$$

Therefore we let

$$
\begin{aligned}
f(B) :=& \operatorname{Tr}\left[\sum_{i<j} (\boldsymbol{b}_i - \boldsymbol{b}_j)(\boldsymbol{b}_i - \boldsymbol{b}_j)^T p_i p_j - \sum_{i<j} (\boldsymbol{b}_i - \boldsymbol{b}_j)(\boldsymbol{b}_i - \boldsymbol{b}_j)^T \frac{1}{n^2}\right] \\
=&\sum_{i<j} \|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2 p_i p_j - \sum_{i<j} \|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2 \frac{1}{n^2} \qquad (\operatorname{Tr}[AB] = \operatorname{Tr}[BA])
\end{aligned}
$$

and use it to compare the trace of covariance matrix of uniform- and nonuniform-subsamplings.

First of all,

$$
\begin{aligned}
&\mathbb{E}[f(B)] \\
=&\mathbb{E}[\|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2] \sum_{i<j} \left(p_i p_j - \frac{1}{n^2}\right) \\
=&\mathbb{E}[\|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2] \left(\sum_{i<j} p_i p_j - \frac{n-1}{2n}\right) \\
=&\mathbb{E}[\|\boldsymbol{b}_i - \boldsymbol{b}_j\|^2] \left(\frac{1 - \sum_{i=1}^n p_i^2}{2} - \frac{n-1}{2n}\right)
\end{aligned}
$$

$$\begin{aligned}&\leq \mathbb{E}[\|\boldsymbol{b}_i-\boldsymbol{b}_j\|^2]\left(\frac{1-\frac{1}{n}}{2}-\frac{n-1}{2n}\right)\\&=0\end{aligned}$$

where the inequality is due to Cauchy-Schwarz and it is a strict inequality unless all p_i's are equal, which means uniform subsampling on average has larger variablity than a non-uniform scheme measured by the trace of covariance matrix.

Moreover, concentration inequality can help show $f(B)$ is negative with high probability if h is small. To this end, plug $\boldsymbol{x}=\mathcal{O}(\sqrt{h})$ in and rewrite

$$p_i=\frac{1}{Z}\exp\left\{Fh\left[\frac{\|\boldsymbol{y}+\frac{1}{n}\sum_{i=1}^n\boldsymbol{b}_i\|^2}{2}-\frac{\|\boldsymbol{y}+\boldsymbol{b}_i\|^2}{2}\right]\right\}$$

where $\boldsymbol{y}=\frac{\sigma}{\sqrt{h}}\boldsymbol{x}=\mathcal{O}(1)$, $F=-\frac{1}{\sigma^2}$ and Z is the normalization constant. Denote the unnormalized probability by

$$\widetilde{p}_i=\exp\left\{Fh\left[\frac{\|\boldsymbol{y}+\frac{1}{n}\sum_{i=1}^n\boldsymbol{b}_i\|^2}{2}-\frac{\|\boldsymbol{y}+\boldsymbol{b}_i\|^2}{2}\right]\right\}$$

and we have

$$\begin{aligned}f(B)&=\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\|\boldsymbol{b}_i-\boldsymbol{b}_j\|^2\left(p_ip_j-\frac{1}{n^2}\right)\\&=\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\|\boldsymbol{b}_i-\boldsymbol{b}_j\|^2\frac{\widetilde{p}_i\widetilde{p}_j}{[\sum_{k=1}^n\widetilde{p}_k]^2}-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\|\boldsymbol{b}_i-\boldsymbol{b}_j\|^2\frac{1}{n^2}\end{aligned}$$

To prove concentration results, it is useful to estimate

$$\begin{aligned}C_i=\sup_{\substack{\boldsymbol{b}_1,\cdots,\boldsymbol{b}_n\in B(\mathbf{0},R)\\ \widehat{\boldsymbol{b}}_i\in B(\mathbf{0},R)}}&|f(\boldsymbol{b}_1,\cdots,\boldsymbol{b}_i,\cdots,\boldsymbol{b}_n)\\&-f(\boldsymbol{b}_1,\cdots,\widehat{\boldsymbol{b}}_i,\cdots,\boldsymbol{b}_n)|\end{aligned}$$

where $B(\mathbf{0},R)$ is a ball centered at origin with radius R in $\mathbb{R}^d$.

Due to the mean value theorem, we have $C_i\leq 2R\sup|\frac{\partial f}{\partial\boldsymbol{b}_i}|$. By symmetry, it suffices to compute $\sup|\frac{\partial f}{\partial\boldsymbol{b}_1}|$ to upper bound C_1. Note that

$$\frac{\partial\widetilde{p}_j}{\partial\boldsymbol{b}_1}=2\widetilde{p}_jFh[\frac{1}{n}(\boldsymbol{y}+\frac{1}{n}\sum_{i=1}^n\boldsymbol{b}_i)-(\boldsymbol{y}+\boldsymbol{b}_j)\delta_{1j}]=\mathcal{O}(h)\widetilde{p}_j$$

where δ_{1j} is the Kronecker delta function. Thus

$$\begin{aligned}&\frac{\partial f}{\partial\boldsymbol{b}_1}=\sum_{j=1}(\boldsymbol{b}_1-\boldsymbol{b}_j)\frac{\widetilde{p}_1\widetilde{p}_j}{[\sum_{k=1}^n\widetilde{p}_k]^2}-\sum_{j=1}^n(\boldsymbol{b}_1-\boldsymbol{b}_j)\frac{1}{n^2}+\sum_{i,j=1}^n\|\boldsymbol{b}_1-\boldsymbol{b}_j\|^2\frac{\mathcal{O}(h)\widetilde{p}_i\widetilde{p}_j}{[\sum_{k=1}^n\widetilde{p}_k]^2}\\&-2\sum_{i,j=1}^n\|\boldsymbol{b}_1-\boldsymbol{b}_j\|^2\frac{\widetilde{p}_i\widetilde{p}_j}{[\sum_{k=1}^n\widetilde{p}_k]^3}\sum_{k=1}^n\widetilde{p}_k\mathcal{O}(h)\\&=\widetilde{p}_1\sum_{j=1}^n(\boldsymbol{b}_1-\boldsymbol{b}_j)\frac{\widetilde{p}_j}{[\sum_{k=1}^n\widetilde{p}_k]^2}-\sum_{j=1}^n(\boldsymbol{b}_1-\boldsymbol{b}_j)\frac{1}{n^2}+\frac{\mathcal{O}(n^2)\mathcal{O}(h)}{\mathcal{O}(n^2)}+\frac{\mathcal{O}(n^2)}{\mathcal{O}(n^3)}\mathcal{O}(n)\mathcal{O}(h)\\&=\mathcal{O}(\frac{h}{n})+\mathcal{O}(h)+\mathcal{O}(h)\\&=\mathcal{O}(h)\end{aligned}$$

where $\mathcal{O}(\frac{h}{n})$ in the 2nd last equation comes from the difference of the first two terms in the 3rd last equation. This estimation shows that $C_i \leq 2R\mathcal{O}(h) = \mathcal{O}(h)$.

Therefore, by McDiarmid's inequality, we conclude for any $\epsilon > 0$,

$$\mathbb{P}(|f - \mathbb{E}[f]| > \epsilon) \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n C_i^2}\right) = 2\exp\left(\frac{-2\epsilon^2}{n\mathcal{O}(h^2)}\right).$$

Any choice of $h(n) = o(n^{-1/2})$ will render this probability asymptotically vanishing as n grows, which means that f will be negative with high probability, which is equivalent to reduced variance per step. □

Appendix C. Proof of Theorem 4.3.

Proof. We rewrite the generator of underdamped Langevin with full gradient as

$$\mathcal{L}f(\boldsymbol{X}) = \boldsymbol{F}(\boldsymbol{X})^T \begin{bmatrix}\nabla_{\boldsymbol{\theta}} f(\boldsymbol{X}) \\ \nabla_{\boldsymbol{r}} f(\boldsymbol{X})\end{bmatrix} + \frac{1}{2} A : \nabla\nabla f(\boldsymbol{X})$$

where

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix}\boldsymbol{r} \\ -\gamma \boldsymbol{r} - \nabla V(\boldsymbol{\theta})\end{bmatrix}, \quad A = GG^T \text{and} \quad G = \begin{bmatrix}O_{d\times d} & O_{d\times d} \\ O_{d\times d} & \sqrt{2\gamma} I_{d\times d}\end{bmatrix}$$

Rewrite the discretized underdamped Langevin with stochastic gradient in variable $\boldsymbol{X}$, i.e.,

$$\boldsymbol{X}_{k+1}^E - \boldsymbol{X}_k^E = h\boldsymbol{F}_k(\boldsymbol{X}_k^E) + \sqrt{h} G_k \boldsymbol{\eta}_{k+1}$$

where

$$\boldsymbol{F}_k(\boldsymbol{X}) = \begin{bmatrix}\boldsymbol{r} \\ -\gamma \boldsymbol{r} - n\nabla V_{I_k}(\boldsymbol{\theta})\end{bmatrix}, \qquad G_k = G = \begin{bmatrix}O_{d\times d} & O_{d\times d} \\ O_{d\times d} & \sqrt{2\gamma} I_{d\times d}\end{bmatrix},$$

and $\boldsymbol{\eta}_{k+1}$ is a $2d$ dimensional standard Gaussian random vector. Note that this representation include both SGHMC and EWSG, for SGHMC I_k follows uniform distribution and for EWSG, I_k follows the MCMC-approximated exponentially weighted distribution.

Denote the generator associated with stochastic gradient underdamped Langevin at the k-th iteration by

$$\mathcal{L}_k f(\boldsymbol{X}) = \boldsymbol{F}_k(\boldsymbol{X})^T \begin{bmatrix}\nabla_{\boldsymbol{\theta}} f(\boldsymbol{X}) \\ \nabla_{\boldsymbol{r}} f(\boldsymbol{X})\end{bmatrix} + \frac{1}{2} A : \nabla\nabla f(\boldsymbol{X})$$

and the difference of the generators of full gradient and stochastic gradient underdamped Langevin at k-th interation is denoted by

$$\begin{aligned}\Delta\mathcal{L}_k f(\boldsymbol{X}) &= (\mathcal{L}_k - \mathcal{L}) f(\boldsymbol{X}) = (\boldsymbol{F}_k(\boldsymbol{X}) - \boldsymbol{F}(\boldsymbol{X}))^T \left[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{X}) \nabla_{\boldsymbol{r}} f(\boldsymbol{X})\right] \\ &= \langle \nabla V(\boldsymbol{\theta}) - n\nabla V_{I_k}(\boldsymbol{\theta}), \nabla_{\boldsymbol{r}} f(\boldsymbol{X})\rangle\end{aligned}$$

For brevity, we write $\phi_k = \phi(\boldsymbol{X}_k^E)$, $\boldsymbol{F}_k^E = \boldsymbol{F}_k(\boldsymbol{X}_k^E)$, $\psi_k = \psi(\boldsymbol{X}_k^E)$ and $D^l\phi_k = (D^l\psi)(\boldsymbol{X}_k^E)$ where $(D^l\psi)(z)$ is the l-th order derivative. We write $(D^l\psi)[\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_l]$ for derivative evaluated in the direction $\boldsymbol{s}_j, j = 1, 2, \cdots, l$. Define

$$\boldsymbol{\delta}_k = \boldsymbol{X}_{k+1}^E - \boldsymbol{X}_k^E = h\boldsymbol{F}_k^E + \sqrt{h} G_k \boldsymbol{\eta}_{k+1}$$

Under the assumptions of Theorem 4.3, we show that the vector field F_k^E also has bounded momentum up to p-th order.

Lemma C.1. *Under the assumption of Theorem 4.3, there exists a constant M such that up to $\frac{p}{2}$-th order moments of random vector field $\boldsymbol{F}_k^E$ are bounded*

$$\mathbb{E}\|\boldsymbol{F}_k^E\|_2^j \leq M,\ \forall j = 0, 1, 2, \cdots, \frac{p}{2},\ \forall k = 0, 1, 2 \cdots,$$

Proof. It suffices to bound the highest moment, as all other lower order moments are bounded by the highest one by Holder's inequality.

First notice that

$$\|\boldsymbol{F}_k^E\|_2 = \left\| \begin{bmatrix} \boldsymbol{r}_k^E \\ -\gamma \boldsymbol{r}_k^E - \nabla V_{I_k}(\boldsymbol{\theta}_k^E) \end{bmatrix} \right\|_2 \leq \sqrt{1+\gamma^2}\|\boldsymbol{r}_k^E\|_2 + \|\nabla V_{I_k}(\theta_k^E)\|_2$$

Hence

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{F}_k^E\|_2^{\frac{p}{2}} \leq & \mathbb{E}\left(\sqrt{1+\gamma^2}\|\boldsymbol{r}_k^E\|_2 + \|\nabla V_{I_k}(\theta_k^E)\|_2\right)^{\frac{p}{2}} \\
= & \mathbb{E}\left\{\sum_{i=0}^{\frac{p}{2}} \binom{\frac{p}{2}}{i} \|\boldsymbol{r}_k^E\|_2^i \|\nabla V_{I_k}(\theta_k^E)\|_2^{\frac{p}{2}-i}\right\} \\
= & \sum_{i=0}^{\frac{p}{2}} \binom{\frac{p}{2}}{i} \mathbb{E}\left[\|\nabla V_{I_k}(\theta_k^E)\|_2^{\frac{p}{2}-i} \|\boldsymbol{r}_k^E\|_2^i\right] \\
\leq & \sum_{i=0}^{\frac{p}{2}} \binom{\frac{p}{2}}{i} \sqrt{\mathbb{E}\left[\|\nabla V_{I_k}(\theta_k^E)\|_2^{p-2i}\right]} \sqrt{\mathbb{E}\left[\|\boldsymbol{r}_k^E\|_2^{2i}\right]} \quad \text{(Cauchy-Schwarz inequality)}
\end{aligned}$$

By assumption, we know each $\mathbb{E}\left[\|\nabla V_{I_k}(\theta_k^E)\|_2^l\right], \mathbb{E}\|\boldsymbol{r}_k^E\|_2^l, l = 0, 1, \cdots, p$ is bounded, so we conclude there exists a constant $M > 0$ that bounds the $\frac{p}{2}$-th order moment of $\boldsymbol{F}_k^E, \forall k = 0, 1, \cdots,$ □

Using Taylor's expansion for ψ, we have

$$\psi_{k+1} = \psi_k + D\psi_k[\boldsymbol{\delta}_k] + \frac{1}{2}D^2\psi_k[\boldsymbol{\delta}_k, \boldsymbol{\delta}_k] + \frac{1}{6}D^3\psi_k[\boldsymbol{\delta}_k, \boldsymbol{\delta}_k, \boldsymbol{\delta}_k] + R_{k+1}$$

where

$$R_{k+1} = \left(\frac{1}{6}\int_0^1 s^3 D^4\psi(s\boldsymbol{X}_k^E + (1-s)\boldsymbol{X}_{k+1}^E)ds\right)[\boldsymbol{\delta}_k, \boldsymbol{\delta}_k, \boldsymbol{\delta}_k, \boldsymbol{\delta}_k]$$

is the remainder term. Therefore, we have

$$\begin{aligned}
\psi_{k+1} = & \psi_k + h\mathcal{L}_k\psi_k + h^{\frac{1}{2}} D\psi_k[G_k\boldsymbol{\eta}_{k+1}] + h^{\frac{3}{2}} D^2\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}] \\
& + \frac{1}{2}h^2 D^2\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E] + \frac{1}{6}D^3\psi_k[\boldsymbol{\delta}_k, \boldsymbol{\delta}_k, \boldsymbol{\delta}_k] + r_{k+1} + R_{k+1}
\end{aligned} \tag{7}$$

where

$$r_{k+1} = \frac{h}{2}\left(D^2\psi_k[G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}] - A : \nabla\nabla\psi_k\right)$$

Summing Equation (7) ove the first K terms, dividing by Kh and use Poisson equation, we have

$$\frac{1}{Kh}(\psi_K - \psi_0) = \frac{1}{K}\sum_{k=0}^{K-1}(\phi_k - \bar{\phi}) + \frac{1}{K}\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k + \frac{1}{Kh}\sum_{i=1}^{3}(M_{i,K} + S_{i,K}), \tag{8}$$

where

$$M_{1,K} = \sum_{k=0}^{K-1} r_{k+1},\ M_{2,K} = h^{\frac{1}{2}} \sum_{k=0}^{K-1} D\psi_k[G_k\boldsymbol{\eta}_{k+1}],\ M_{3,K} = h^{\frac{3}{2}} \sum_{k=0}^{K-1} D^2\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}],$$
$$S_{1,K} = \frac{h^2}{2} \sum_{k=0}^{K-1} D^2\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E],\ S_{2,K} = \sum_{k=0}^{K-1} R_{k+1},\ S_{3,K} = \frac{1}{6} \sum_{k=0}^{K-1} D^3\psi_k[\boldsymbol{\delta}_k, \boldsymbol{\delta}_k, \boldsymbol{\delta}_k]$$

Furthermore, it will be convenient to decompose

$$S_{3,K} = M_{0,K} + S_{0,K}$$

where

$$S_{0,K} = h^2 \sum_{k=0}^{K-1} \left(hD^3\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E, \boldsymbol{F}_k^E] + 3D^3\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}] \right)$$
$$M_{0,K} = h^{\frac{3}{2}} \sum_{k=0}^{K-1} \left(D^3\psi_k[G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}] + 3hD^3\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}] \right)$$

Rearrange terms in Equation (7), square on both sides, use Cauchy-Schwarz inequality and take expectation, we have

$$\begin{aligned}
&\mathbb{E}\big(\hat{\phi}_K - \bar{\phi}\big)^2 \\
\leq & C \left[\mathbb{E}\frac{(\psi_K - \psi_0)^2}{(Kh)^2} + \frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1} (\Delta\mathcal{L}_k\psi_k) \right)^2 \right. \\
& \left. + \frac{1}{(Kh)^2} \sum_{i=0}^{2} \mathbb{E}S_{i,K}^2 + \frac{1}{(Kh)^2} \sum_{i=0}^{3} \mathbb{E}M_{i,K}^2 \right] \\
= & C \left[\mathbb{E}\frac{(\psi_K - \psi_0)^2}{T^2} + \frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1} (\Delta\mathcal{L}_k\psi_k) \right)^2 + \frac{1}{T^2} \sum_{i=0}^{2} \mathbb{E}S_{i,K}^2 + \frac{1}{T^2} \sum_{i=0}^{3} \mathbb{E}M_{i,K}^2 \right]
\end{aligned} \tag{9}$$

where $T = kh$, the corresponding time of the underlying continuous dynamics.

We now show how each term is bounded. By boundedness of ψ, we have

$$\mathbb{E}\frac{(\psi_K - \psi_0)^2}{T^2} \leq \frac{4\|\psi\|_\infty^2}{T^2} = \mathcal{O}(\frac{1}{T^2})$$

The second term $\frac{1}{K^2}\mathbb{E}\big(\sum_{k=0}^{K-1}(\Delta\mathcal{L}_k\psi_k)\big)^2$ is critical in showing the advantage of EWSG, and we will show how to derive its bound in detail later.

The technique we use to bound $\frac{1}{T^2}\mathbb{E}S_{i,K}^2, i = 0, 1, 2$ are all similar, we will first show an upper bound for $|S_{i,K}|$ in terms of powers of $\|\boldsymbol{F}_k^E\|$, then take square and expectation, and finally expand squares and use Lemma C.1 extensively to derive bounds. As a concrete example, we will show how to bound $\frac{1}{T^2}\mathbb{E}S_{0,K}^2$. Other bounds follow in a similar fashion and details are omitted.

To bound the term containing $S_{0,K}$, we first note that

$$\begin{aligned}|S_{0,K}| \leq & h^2 \sum_{k=0}^{K-1} \left(h|D^3\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E, \boldsymbol{F}_k^E]| + 3|D^3\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}]|\right) \\ \leq & h^2\|D^3\psi\|_\infty \sum_{k=0}^{K-1} \left(h\|\boldsymbol{F}_k^E\|_2^3 + 3\|\boldsymbol{F}_k^E\|_2\|G_k\boldsymbol{\eta}_{k+1}\|_2^2\right)\end{aligned}$$

Square both sides of the above inequality and take expectation, we obtain

$$\begin{aligned}&\frac{1}{T^2}\mathbb{E}|S_{0,K}|^2 \qquad (10)\\ \leq & \frac{h^4}{T^2}\|D^3\psi\|_\infty^2 \mathbb{E}\big(\sum_{k=0}^{K-1} h\|\boldsymbol{F}_k^E\|_2^3 + 3\|\boldsymbol{F}_k^E\|_2\|G_k\boldsymbol{\eta}_{k+1}\|_2^2\big)^2 \\ \leq & \frac{h^4}{T^2}\|D^3\psi\|_\infty^2 K \sum_{k=0}^{K-1} \mathbb{E}(h\|\boldsymbol{F}_k^E\|_2^3 + 3\|\boldsymbol{F}_k^E\|_2\|G_k\boldsymbol{\eta}_{k+1}\|_2^2)^2 \\ & \text{(Cauchy-Schwarz inequality)} \\ = & \frac{h^4}{T^2}\|D^3\psi\|_\infty^2 K \sum_{k=0}^{K-1} \mathbb{E}[h^2\|\boldsymbol{F}_k^E\|_2^6 + 6\|\boldsymbol{F}_k^E\|_2^4\|G_k\boldsymbol{\eta}_{k+1}\|_2^2 + 9\|\boldsymbol{F}_k^E\|_2^2\|G_k\boldsymbol{\eta}_{k+1}\|_4^2] \\ = & \frac{h^4}{T^2}\|D^3\psi\|_\infty^2 K \sum_{k=0}^{K-1} h^2\mathbb{E}\|\boldsymbol{F}_k^E\|_2^6 + 6\mathbb{E}\|\boldsymbol{F}_k^E\|_2^4\mathbb{E}\|G_k\boldsymbol{\eta}_{k+1}\|_2^2 + 9\mathbb{E}\|\boldsymbol{F}_k^E\|_2^2\mathbb{E}\|G_k\boldsymbol{\eta}_{k+1}\|_4^2 \\ = & \frac{1}{T^2}\mathcal{O}(K^2h^4) \\ = & \mathcal{O}(h^2)\end{aligned}$$

To bound the term containing $S_{1,K}$ and $S_{2,K}$, we have

$$\begin{aligned}|S_{1,K}| \leq & \frac{h^2}{2}\sum_{k=0}^{K-1}\|D^2\psi\|_\infty\|\boldsymbol{F}_k^E\|_2^2 \\ |S_{2,K}| \leq & \frac{1}{24}\|D^4\psi\|_\infty \sum_{k=0}^{K-1}\|\boldsymbol{\delta}_k\|_2^4 \\ \leq & \frac{1}{24}h^2\|D^4\psi\|_\infty \sum_{k=0}^{K-1}\|\sqrt{h}\boldsymbol{F}_k^E + G_k\boldsymbol{\eta}_{k+1}\|_2^4\end{aligned}$$

Then we can obtain the following bound in a similar fashion as in Equation (10)

$$\begin{aligned}\frac{1}{T^2}\mathbb{E}S_{1,K}^2 = & \mathcal{O}(h^2) \\ \frac{1}{T^2}\mathbb{E}S_{2,K}^2 = & \mathcal{O}(h^2)\end{aligned}$$

Now we will use martingale argument to bound $\frac{1}{T^2}\mathbb{E}M_{i,K}^2, i = 0, 1, 2, 3$. There are two injected randomness at k-th iteration, the Gaussian noise $\boldsymbol{\eta}_{k+1}$ and the stochastic gradient term determined by the stochastic index I_k. Denote the sigma algebra at k-th iteration by $\mathcal{F}_k$. For both SGHMC and EWSG we have

$$\boldsymbol{\eta}_{k+1} \perp \mathcal{F}_k \text{ and } I_k \perp \boldsymbol{\eta}_{k+1}$$

hence

$$\begin{aligned}\mathbb{E}[\boldsymbol{\eta}_{k+1}|\mathcal{F}_k] =&\mathbf{0}\\ \mathbb{E}[D^3\psi_k[G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}]|\mathcal{F}_k] =&0\\ \mathbb{E}[D^2\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}]|\mathcal{F}_k] =&0\\ \mathbb{E}[D^3\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}]|\mathcal{F}_k] =&0\end{aligned}$$

Therefore, it is clear that $M_{i,K}, i = 0, 1, 2, 3$ are all martingales. Due to martingale properties, we have

$$\begin{aligned}&\frac{1}{T^2}\mathbb{E}M_{0,K}^2\\ &= \frac{h^3}{T^2}\sum_{k=0}^{K-1}\mathbb{E}\big(D^3\psi_k[G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}, G_k\boldsymbol{\eta}_{k+1}] + 3hD^3\psi_k[\boldsymbol{F}_k^E, \boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}]\big)^2\\ &\qquad =\frac{1}{T^2}\mathcal{O}(h^3K) = \mathcal{O}(\frac{h^2}{T})\\ &\frac{1}{T^2}\mathbb{E}M_{1,K}^2 =\frac{1}{T^2}\sum_{k=0}^{K-1}\mathbb{E}r_{k+1}^2 = \frac{1}{T^2}\mathcal{O}(h^2K) = \mathcal{O}(\frac{h}{T})\\ &\frac{1}{T^2}\mathbb{E}M_{2,K}^2 =\frac{h}{T^2}\sum_{k=0}^{K-1}\mathbb{E}(D\psi_k[G_k\boldsymbol{\eta}_{k+1}])^2 = \frac{1}{T^2}\mathcal{O}(hK) = \mathcal{O}(\frac{1}{T})\\ &\frac{1}{T^2}\mathbb{E}M_{3,K}^2 =\frac{1}{T^2}h^3\sum_{k=0}^{K-1}\mathbb{E}(D^2\psi_k[\boldsymbol{F}_k^E, G_k\boldsymbol{\eta}_{k+1}])^2 = \frac{1}{T^2}\mathcal{O}(h^3K) = \mathcal{O}(\frac{h^2}{T})\end{aligned}$$

We now collect all bounds derived so far and obtain

$$\begin{aligned}&\mathbb{E}\big(\hat{\phi}_K - \bar{\phi}\big)^2\\ \leq& C\left[\mathcal{O}(\frac{1}{T^2}) + \frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}(\Delta\mathcal{L}_k\psi_k)\right)^2 + \mathcal{O}(h^2) + \mathcal{O}(\frac{h}{T}) + \mathcal{O}(\frac{1}{T}) + \mathcal{O}(\frac{h^2}{T})\right]\\ \leq& C\left[\mathcal{O}(\frac{1}{T}) + \frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}(\Delta\mathcal{L}_k\psi_k)\right)^2 + \mathcal{O}(h^2)\right]\end{aligned} \tag{11}$$

In the above inequality, we use $\frac{1}{T^2} < \frac{1}{T}$ and $\frac{h}{T} \leq \frac{1}{T}$, $\frac{h^2}{T} \leq \frac{1}{T}$ as typically we assume $T \gg 1$ and $h \ll 1$ in non-asymptotic analysis.

Now we focus on the remaining term $\frac{1}{K^2}\mathbb{E}\big(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\big)^2$. For SGHMC, we have that $\mathbb{E}[\Delta\mathcal{L}_k\psi_k|\mathcal{F}_k] = 0$, hence $\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k$ is a martingale. By martingale property, we have

$$\frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\right)^2 = \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2$$

For EWSG, $\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k$ is no longer a martingale, but we still have the following

$$\frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\right)^2 =\frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathbb{E}(\Delta\mathcal{L}_i\psi_i)(\Delta\mathcal{L}_j\psi_j)$$

$$= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathbb{E}[(\Delta\mathcal{L}_i\psi_i)\mathbb{E}[\Delta\mathcal{L}_j\psi_j|\mathcal{F}_j]] \tag{12}$$

For the term $\mathbb{E}[\Delta\mathcal{L}_j\psi_j|\mathcal{F}_j]$, we have

$$\begin{aligned}\mathbb{E}[\Delta\mathcal{L}_j\psi_j|\mathcal{F}_j] &= \mathbb{E}[\langle\nabla V(\boldsymbol{\theta}_j^E) - n\nabla V_{I_j}(\boldsymbol{\theta}_j^E), \nabla_{\boldsymbol{r}}\psi_j\rangle|\mathcal{F}_j] \\ &= \langle\mathbb{E}[\nabla V(\boldsymbol{\theta}_j^E) - n\nabla V_{I_j}(\boldsymbol{\theta}_j^E)|\mathcal{F}_j], \nabla_{\boldsymbol{r}}\psi_j\rangle\end{aligned}$$

as $\psi_j \in \mathcal{F}_j$. Then by Cauchy-Schwarz inequality, boundedness of ψ and the fact $\|\nabla V(\boldsymbol{\theta}_j^E) - \mathbb{E}[n\nabla V_{I_j}(\boldsymbol{\theta}_j^E)|\mathcal{F}_j]\|_2 = \mathcal{O}(h)$ as shown in the proof of Theorem 4.2, we conclude $\mathbb{E}[\Delta\mathcal{L}_j\psi_j|\mathcal{F}_j] = \mathcal{O}(h)$.

Now plug the above result in Equation (12), we have

$$\begin{aligned}\frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\right)^2 &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathbb{E}[(\Delta\mathcal{L}_i\psi_i)\mathbb{E}[\Delta\mathcal{L}_j\psi_j|\mathcal{F}_j]] \\ &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathbb{E}[\Delta\mathcal{L}_i\psi_i]\mathcal{O}(h) \\ &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathcal{O}(h^2) \\ &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \frac{2}{K^2}\sum_{i<j}\mathcal{O}(h^2) \\ &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \mathcal{O}(h^2)\end{aligned}$$

Combine both cases of SGHMC and EWSG, we obtain

$$\frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\right)^2 = \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 + \mathcal{O}(h^2)$$

Note that $\mathcal{O}(h^2)$ term will later be combined with other error terms with the same order.

The final piece is to bound $\frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2$, and we have

$$\begin{aligned}\frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}(\Delta\mathcal{L}_k\psi_k)^2 &= \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}\langle\nabla V(\boldsymbol{\theta}_k^E) - n\nabla V_{I_k}(\boldsymbol{\theta}_k^E), \nabla_{\boldsymbol{r}}\psi_k\rangle^2 \\ &\le \frac{1}{K^2}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla V(\boldsymbol{\theta}_k^E) - n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\|_2^2 \cdot \|\nabla_{\boldsymbol{r}}\psi_k\|_2^2] \\ &\quad \text{(Cauchy-Schwarz inequality)} \\ &\le \frac{M_3^2}{K^2}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla V(\boldsymbol{\theta}_k^E) - n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\|_2^2] \\ &= \frac{M_3^2}{K^2}\sum_{k=0}^{K-1}\mathbb{E}[\mathbb{E}[\|\nabla V(\boldsymbol{\theta}_k^E) - n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\|_2^2 \,|\, \mathcal{F}_k]]\end{aligned}$$

$$\leq \frac{2M_3^2}{K^2}\sum_{k=0}^{K-1}\mathbb{E}[\underbrace{\mathbb{E}[\,\|\nabla V(\boldsymbol{\theta}_k^E)-\mathbb{E}[n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\,|\,\mathcal{F}_k]\|^2\,|\,\mathcal{F}_k]}_{Q_1}]$$
$$+\underbrace{\mathbb{E}[\,\|\mathbb{E}[n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\,|\,\mathcal{F}_k]-n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\|_2^2\,|\,\mathcal{F}_k]}_{Q_2}]$$

The term Q_1 captures the bias of stochastic gradient. For SGHMC, uniform gradient subsamping leads to an unbiased gradient estimator, so $Q_1 = 0$ for SGHMC. For EWSG, same as in the proof of Theorem 2, we have that

$$\mathbb{E}\left[\,\|\nabla V(\boldsymbol{\theta}_k^E)-\mathbb{E}[n\nabla V_{I_k}(\boldsymbol{\theta}_k^E)\,|\,\mathcal{F}_k]\|^2\,|\,\mathcal{F}_k\right]=\mathcal{O}(h^2)$$

Combining two cases, we have

$$Q_1=\mathcal{O}(h^2)$$

For a random vector $\boldsymbol{v}$ with mean $\mathbb{E}[\boldsymbol{v}]=\mathbf{0}$, we have

$$\mathbb{E}[\|\boldsymbol{v}\|^2]=\mathbb{E}\left[\mathrm{Tr}[\boldsymbol{v}\boldsymbol{v}^T]\right]=\mathrm{Tr}\left[\mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T]\right]=\mathrm{Tr}\left[\mathrm{cov}(\boldsymbol{v})\right]$$

where $\mathrm{cov}(\boldsymbol{v})$ is the covariance matrix of random vector $\boldsymbol{v}$. Therefore, we have that

$$Q_2=\mathrm{Tr}\left[\mathrm{cov}(n\nabla V_{I_k}|\mathcal{F}_k)\right],$$

i.e., Q_2 is the trace of the covariance matrix of stochastic gradient estimate conditioned on current filtration $\mathcal{F}_k$.

Combining Q_1 and Q_2, we have that

$$\frac{1}{K^2}\mathbb{E}\left(\sum_{k=0}^{K-1}\Delta\mathcal{L}_k\psi_k\right)^2\leq\frac{2M_3^2}{K^2}\sum_{k=0}^{K-1}\left[\mathbb{E}[\mathrm{Tr}[\mathrm{cov}(n\nabla V_{I_k}|\mathcal{F}_k)]]+\mathcal{O}(h^2)\right]$$
$$=\frac{2M_3^2h}{T}\frac{\sum_{k=0}^{K-1}\mathbb{E}[\mathrm{Tr}[\mathrm{cov}(n\nabla V_{I_k}|\mathcal{F}_k)]]}{K}+\mathcal{O}(\frac{h^3}{T})$$

Now plug this bound into Equation (11) and we obtain

$$\mathbb{E}\big(\hat{\phi}_K-\bar{\phi}\big)^2\leq C_1\frac{1}{T}+C_2\frac{h}{T}\frac{\sum_{k=0}^{K-1}\mathbb{E}\left[\mathrm{Tr}[\mathrm{cov}(n\nabla V_{I_k}|\mathcal{F}_k)]\right]}{K}+C_3h^2$$

for some constants $C_1, C_2, C_3 > 0$ depending on M_1, M_2, M_3. □

Appendix D. **Mini batch version of EWSG.** When mini batch size $b > 1$, for each mini batch $\{i_1, i_2, \cdots, i_b\}$, we use $\frac{n}{b}\sum_{j=1}^{b}\nabla V_{i_j}$ to approximate full gradient ∇V, and assign the mini batch $\{i_1, i_2, \cdots, i_b\}$ probability $p_{i_1 i_2,\cdots,i_b}$. We can easily extend the transition probability of $b = 1$ to general b, simply by replacing $n\nabla V_i$ with $\frac{n}{b}\sum_{j=1}^{b}\nabla V_{i_j}$ and end up with

$$\tilde{P}(\boldsymbol{\theta}_{k+1},\boldsymbol{r}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{r}_k)=\delta(\boldsymbol{\theta}_{k+1}=\boldsymbol{\theta}_k+\boldsymbol{r}_k h)\times$$
$$\sum_{i_1,i_2,\cdots,i_b}p_{i_1 i_2\cdots i_b}\Phi\left(\boldsymbol{x}+n\boldsymbol{a}_{i_1 i_2\cdots i_b}\right)\frac{1}{\sigma\sqrt{h}}$$

where

$$\boldsymbol{x}=\frac{\boldsymbol{r}_{k+1}-\boldsymbol{r}_k+h\gamma\boldsymbol{r}_k}{\sigma\sqrt{h}},\quad \boldsymbol{a}_{i_1 i_2\cdots i_b}=\frac{\sqrt{h}}{\sigma}\frac{1}{b}\sum_{j=1}^{b}\nabla V_{i_j}(\boldsymbol{\theta}_k)$$

Therefore, to match the transition probability of underdamped Langevin dynamics with stochastic gradient and full gradient, we let $p_{i_1 i_2 \cdots i_b} =$

$$\frac{1}{Z} \exp\left\{ \frac{1}{2} \left[\|\boldsymbol{x} + n\boldsymbol{a}_{i_1 i_2 \cdots i_b}\|^2 - \|\boldsymbol{x} + \sum_{i_1 i_2 \cdots i_b} \boldsymbol{a}_{i_1 i_2 \cdots i_b}\|^2 \right] \right\}$$

where Z is a normalization constant.

To sample multidimensional random data indices $I_1, \cdots, I_b$ from $p_{i_1 i_2 \cdots i_b}$, we again use a Metropolis chain, whose acceptance probability only depends on $a_{i_1 i_2 \cdots i_b}$ and $a_{j_1 j_2 \cdots j_b}$ but not the full gradient.

Appendix E. **EWSG version for overdamped Langevin.** Overdamped Langevin equation is the following SDE

$$d\boldsymbol{\theta}_t = -\nabla V(\boldsymbol{\theta}_t)dt + \sqrt{2} d\boldsymbol{B}_t$$

where $V(\boldsymbol{\theta}) = \sum_{i=1}^n V_i(\boldsymbol{\theta})$ and B_t is a d-dimensional Brownian motion. The Euler-Maruyama discretization is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - h\nabla V(\boldsymbol{\theta}_k) + \sqrt{2h}\boldsymbol{\xi}_{k+1}$$

where $\boldsymbol{\xi}_{k+1}$ is a d-dimensional random Gaussian vector. When stochastic gradient is used, the above numerical schedme turns to

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - h\nabla V_{I_k}(\boldsymbol{\theta}_k) + \sqrt{2h}\boldsymbol{\xi}_{k+1}$$

where I_k is the datum index used in k-th iteration to estimate the full gradient.

Denote $\boldsymbol{x} = \frac{\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k}{\sqrt{2h}}$ and $\boldsymbol{a}_i = \frac{\sqrt{h}\nabla V_i(\boldsymbol{\theta}_k)}{\sqrt{2}}$. If we set

$$p_i = \mathbb{P}(I_k = i) \propto \exp\Big\{ - \frac{\|\boldsymbol{x} + \sum_{j=1}^n \boldsymbol{a}_j\|^2}{2} + \frac{\|\boldsymbol{x} + n\boldsymbol{a}_i\|^2}{2} \Big\}$$

and follow the same steps in Sec.4.2, we will see the transition kernel of the full gradient method being approximated by that of the stochastic gradient version.

Appendix F. **Variance reduction (VR).** We have seen that when step size h is large, EWSG still introduces extra variance. To further mitigate this inaccuracy, we provide in this section a complementary variance reduction technique.

Locally (i.e., conditioned on the state of the system at the current step), we have increased variance

$$\begin{aligned} \operatorname{cov}[\boldsymbol{r}_{k+1}|\boldsymbol{r}_k] &= \mathbb{E}[\operatorname{cov}[\boldsymbol{r}_{k+1}|I]] + \operatorname{cov}[\mathbb{E}[\boldsymbol{r}_{k+1}|I]] \\ &= h(\Sigma_{k+1}^2 + h \operatorname{cov}[n\nabla V_I(\boldsymbol{\theta}_k)]) \end{aligned} \tag{13}$$

where $\Sigma_{k+1}^2 = \frac{1}{h}\mathbb{E}[\operatorname{cov}[\boldsymbol{r}_{k+1}|I]]$. The extra randomness due to the randomness of the index I enters the parameter space through the coupling of $\boldsymbol{\theta}$ and $\boldsymbol{r}$ and eventually deviates the stationary distribution from that of the original dynamics. Adopting the perspective of modified equation [5, 31, 26], we model this as an enlarged diffusion coefficient. To correct for this enlargement and still sample from the correct distribution, we can either, in each step, shrink the size of intrinsic noise to $\Sigma_k \in \mathbb{R}^{d\times d}$ such that $\sigma^2 I = \Sigma_k^2 + h\operatorname{cov}[n\nabla V_I(\boldsymbol{\theta}_{k-1})]$, or alternatively increase the dissipation. More precisely, due to the matrix version fluctuation dissipation theorem $\Sigma^2 = 2\Gamma T$, one could instead increase the friction coefficient $\Gamma \in \mathbb{R}^{d\times d}$ rather than shrinking the intrinsic noise. The second approach is computationally

more efficient because it no longer requires square-rooting / Cholesky decomposition of (possibly large-scale) matrices. Therefore, in each step, we set

$$\Gamma_k = \frac{1}{2T}(\sigma^2 I + h\mathrm{cov}[n\nabla V_I(\boldsymbol{\theta}_{k-1})]).$$

Accurately computing $\mathrm{cov}[n\nabla V_I(\boldsymbol{\theta}_{k-1})]$ is expensive as it requires running I through $1, \cdots, n$, which defeats the purpose of introducing a stochastic gradient. To downscale the computation cost from $\mathcal{O}(n)$ to $\mathcal{O}(1)$, we use an SVRG type estimation of the this variance instead. More specifically, we periodically compute $\mathrm{cov}[n\nabla V_I(\boldsymbol{\theta}_{k-1})]$ only every L data passes, in an outer loop. In every iteration of an inner loop, which integrates the Langevin, an estimate of $\mathrm{cov}[n\nabla V_I(\boldsymbol{\theta}_{k-1})]$ is updated in an SVRG fashion. sof See Algorithm 2 for detailed description. We refer variance reduced variant of EWSG as EWSG-VR.

Algorithm 2 EWSG-VR

Input: {number of data terms n, gradient functions $\nabla V_i(\cdot)$, step size h, number of data passes K, period of variance calibration L, index chain length M, friction and noise coefficients γ and σ}
initialize $\boldsymbol{\theta}_0, \boldsymbol{r}_0, \gamma_0 = \gamma$
initialize inner loop index $k = 0$
for $l = 1, 2, \cdots, K$ **do**
  if $(l-1) \mod L = 0$ **then**
    compute $\boldsymbol{m}_1 \leftarrow \mathbb{E}_I[n\nabla V_I(\boldsymbol{\theta}_k)]$, $\boldsymbol{m}_2 \leftarrow \mathbb{E}_I[n^2\nabla V_I(\boldsymbol{\theta}_k)\nabla V_I(\boldsymbol{\theta}_k)^T]$
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\theta}_k$
  else
    for $t = 1, 2, \cdots, \lceil \frac{n}{M+1} \rceil$ **do**
      $i \leftarrow$ uniformly sampled from $1, \cdots, n$, compute and store $n\nabla V_i(\boldsymbol{\theta}_k)$
      for $m = 1, 2, \cdots, M$ **do**
        $j \leftarrow$ uniformly sampled from $1, \cdots, n$, compute and store $n\nabla V_j(\boldsymbol{\theta}_k)$
        $i \leftarrow j$ with probability in Equation 6
      end for
      update $(\boldsymbol{\theta}_{k+1}, \boldsymbol{r}_{k+1}) \leftarrow (\boldsymbol{\theta}_k, \boldsymbol{r}_k)$ according to Equation 3, using $n\nabla V_i(\boldsymbol{\theta}_k)$ as gradient and Γ_k as friction
      $\boldsymbol{m}_1 \leftarrow \boldsymbol{m}_1 + \nabla V_i(\boldsymbol{\theta}_k) - \nabla V_i(\boldsymbol{\omega})$
      $\boldsymbol{m}_2 \leftarrow \boldsymbol{m}_2 + n\nabla V_i(\boldsymbol{\theta}_k)\nabla V_i(\boldsymbol{\theta}_k)^T - n\nabla V_i(\boldsymbol{\omega})\nabla V_i(\boldsymbol{\omega})^T$
      $\mathrm{covar} \leftarrow \boldsymbol{m}_2 - \boldsymbol{m}_1\boldsymbol{m}_1^T$
      $\Gamma_{k+1} \leftarrow \frac{1}{2T}(\sigma^2 \mathrm{I} + h\,\mathrm{covar})$
      $k \leftarrow k + 1$
    end for
  end if
end for

To demonstrate the performance of EWSG-VR, we reuse the setup of simple Gaussian example in subsection 5.1. As shown in Algorithm 2, the only hyper-parameter of EWSG-VR additional to EWSG is the period of variance calibration, for which we set $L = 1$. All other hyper-parameters (e.g. step size h, friction coefficient γ) are set the same as EWSG. We also run underdamped Langevin dynamics with full gradient (FG) using the same hyper-parameters of EWSG. We plot the KL divergence in Figure 4. We see that EWSG-VR further reduces variance and achieves better statistical accuracy measured in KL divergence. Although

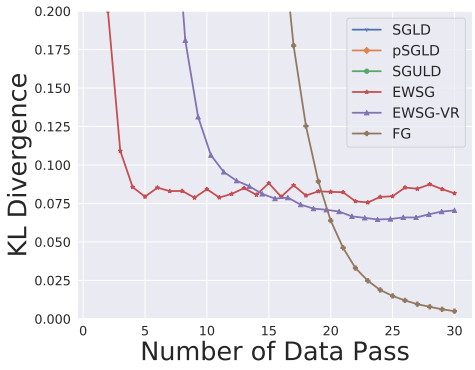
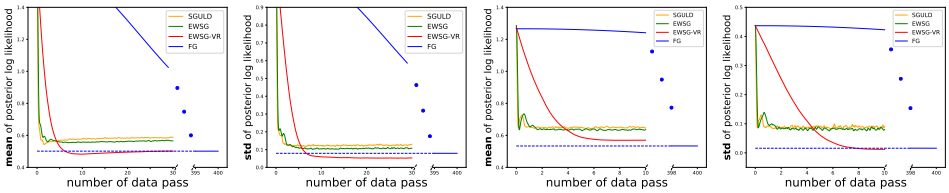


FIGURE 4. KL divergence

EWSG-VR periodically use full data set to calibrate variance estimation, it is still significantly faster than the full gradient version. Note that KL divergence of SGLD, pSGLD and SGHMC are too large so that we can not even see them in Figure 4

We also consider applying EWSG-VR to Bayesian logistic regression problems. We run experiments on two standard classification data sets `parkinsons` [8], `pima`[9] from UCI repository [28].



(A) parkinsons (B) pima

FIGURE 5. Posterior prediction of mean (*left*) and standard deviation (*right*) of log likelihood on test data set generated by SGHMC, EWSG and EWSG-VR on two Bayesian logistic regression tasks. Statistics are computed based on 1000 independent simulations. Minibatch size $b = 1$ for all methods except FG. $M = 1$ for EWSG and EWSG-VR.

From Figure 5, we see stochastic gradient methods (SGHMC, EWSG and EWSG-VR) only take tens of data passes to converge while full gradient version (FG) requires hundreds of data passes to converge. Compared with SGHMC, EWSG produces closer results to FG for which we treat as ground truth, in terms of statistical accuracy. With variance reduction, EWSG-VR is able to achieve even better performance, significantly improving the accuracy of the prediction of mean and standard deviation of log likelihood. It, however, converges slower than EWSG without VR.

One downside of EWSG-VR is that it periodically use whole data set to calibrate variance estimation, so it may not be suitable for very large data sets (e.g.

[8] https://archive.ics.uci.edu/ml/datasets/parkinsons
[9] https://archive.ics.uci.edu/ml/datasets/diabetes

Covertype data set used in subsection 5.2) for which stochastic gradient methods could converge within one data pass.

Appendix G. Additional experiments.

G.1. **A misspecified Gaussian case.** In this subsection, we follow the same setup as in [4] and study a misspecified Gaussian model where one fits a one-dimensional normal distribution $p(\theta) = \mathcal{N}(\theta|\mu_0, \sigma_0^2)$ to 10^5 i.i.d points drawn according to $X_i \sim \log\mathcal{N}(0,1)$, and flat prior is assigned $p(\mu_0, \log\sigma_0) \propto 1$. It was shown in [4] that FlyMC algorithm behaves erratically in this case, as "bright" data points with large values are rarely updated and they drive samples away from the target distribution. Consequently the chain mixes very slowly. One important commonality FlyMC shares with EWSG is that in each iteration, both algorithms select a subset of data in a non-uniform fashion. Therefore, it is interesting to investigate the performance of EWSG in this misspecified model.

For FlyMC[10], a tight lower bound based on Taylor's expansion is used to minimize "bright" data points used per iteration. At each iteration, 10% data points are resampled and turned "on/off" accordingly and the step size is adaptively adjusted. FlyMC algorithm is run for 10000 iterations. Figure 6a shows the histogram of number of data points used in each iteration for FlyMC algorithm. On average, FlyMC consumes 10.9% of all data points per iteration. For fair comparison, the minibatch size of EWSG is hence set $10^5 \times 10.9\% = 10900$ and we run EWSG for 1090 data passes. We set step size $h = 1 \times 10^{-4}$ and friction coefficient $\gamma = 300$ for EWSG. An isotropic random walk Metropolis Hastings (MH) is also run for sufficiently long and serves as the ground truth.

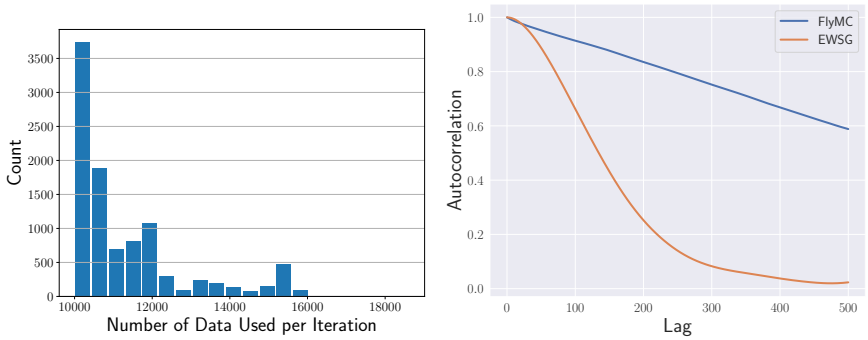
Figure 6b shows the autocorrelation of three algorithms. The autocorrelation of FlyMC decays very slowly, samples that are even 500 iterations away still show strong correlation. The autocorrelation of EWSG, on the other hand, decays much faster, suggesting EWSG explores parameter space efficiently than FlyMC does. Figure 6c and 6d show the samples (the first 1000 samples are discarded as burn-in) generated by EWSG and FlyMC respectively. The samples of EWSG center around the mode of the target distribution while the samples of FlyMC are still far away from the true posterior. The experiment shows EWGS works quite well even in misspecified models, and hence is an effective candidate in combining importance sampling with scalable Bayesian inference.

G.2. **Additional results of BNN experiment.** We report the test error of various SG-MCMC methods after 200 epochs in Table 2. For both MLP and CNN architecture, EWSG outperforms its uniform counterpart SGHMC as well as other benchmarks SGLD, pSGLD and CP-SGHMC. The results clearly demonstrate the effectiveness of the proposed EWSG on deep models.

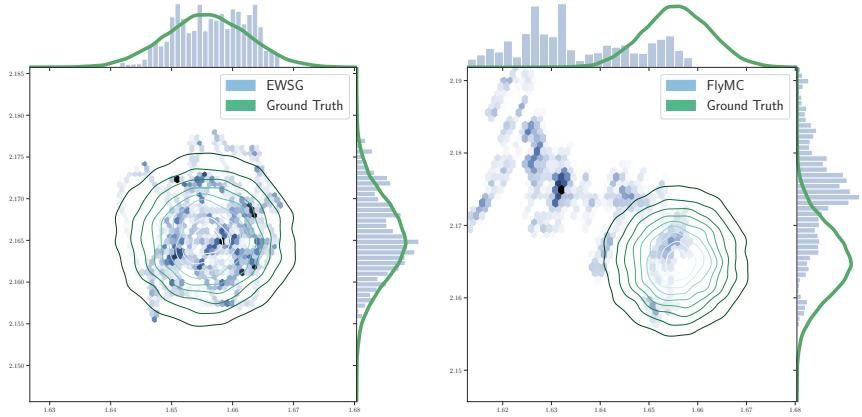
G.3. **Additional experiment on BNN: Tuning M.** In each iteration of EWSG, we run an index Markov chain of length M and select a "good" minibatch to estimate gradient, therefore EWSG essentially uses $b \times (M+1)$ data points per iteration where b is minibatch size. How does EWSG compare with its uniform gradient subsampling counterpart with a larger minibatch size $(b \times (M+1))$?

We empirically answer this question in the context of BNN with MLP architecture. We use the same step size for SGHMC and EWSG and experiment a large

[10] https://github.com/rbardenet/2017-JMLR-MCMCForTallData



(A) Histogram (B) Autocorrelation



(C) Samples of EWSG (D) Samples of FlyMC

FIGURE 6. (a) Histogram of data used in each iteration for FlyMC algorithm. (b) Autocorrelation plot of FlyMC, EWSG and MH. (c) Samples of EWSG. (d) Samples of FlyMC.

TABLE 2. Test error (mean ± standard deviation) after 200 epoches.

Method	Test Error(%), MLP	Test Error(%), CNN
SGLD	1.976 ± 0.055	0.848 ± 0.060
pSGLD	1.821 ± 0.061	0.860 ± 0.052
SGHMC	1.833 ± 0.073	0.778 ± 0.040
CP-SGHMC	1.835 ± 0.047	0.772 ± 0.055
EWSG	**1.793 ± 0.100**	**0.753 ± 0.035**

range of values of minibatch size b and index chain length M. Each algorithm is run for 200 data passes and 10 independent samples are drawn to estimate test error. The results are shown in Table 3. We find that EWSG beats SGHMC with larger minibatch in 8 out of 9 comparison groups, which suggests in general EWSG could be a better way to consuming data compared to increasing minibatch size and may shed light on other areas where stochastic gradient methods are used (e.g. optimization).

b	$M+1=2$	$M+1=5$	$M+1=10$
100	**1.86%** 1.94%	**1.83%** 1.92%	**1.80%** 1.97%
200	1.90% **1.87%**	**1.87%** 1.97%	**1.80%** 2.07%
500	**1.79%** 1.97%	**2.01%** 2.17%	**2.36%** 2.37%

TABLE 3. Test errors of EWSG (top of each cell) and SGHMC (bottom of each cell) after 200 epoches. b is minibatch size for EWSG, and minibatch size of SGHMC is set as $b \times (M+1)$ to ensure the same number of data used per parameter update for both algorithms. Step size is set $h = \frac{10}{b(M+1)}$ as suggested in [12], different from that used to produce Table 2. Results with smaller test error is highlighted in boldface.

Appendix H. EWSG does not necessarily change the speed of convergence significantly. Changing the weights of stochastic gradient from uniform to non-uniform, as we saw, can increase the statistical accuracy of the sampling; however, it does not necessarily increase or decrease the speed of convergence to the (altered) limiting distribution. Numerical examples already demonstrated this fact, but on the theoretical side, we note the non-asymptotic bound provided by Theorem 4.3 may not be tight in terms of the speed of convergence due to its generality. Therefore, here we quantify the convergence speed on a simple quadratic example:

Consider $V_i(\theta) = \frac{1}{n}(\theta - \mu_i)^2/2$ where μ_i's are constant scalars. Assume without loss of generality that $\sum_i \mu_i = 0$, and thus $V(\theta) = \sum_{i=1}^n V_i(\theta) = \theta^2/2 +$ some constant. We will show the convergence speed of $\mathbb{E}\theta$ is comparable for uniform and a class of non-uniform SG-MCMC (including EWSG) applied to second-order Langevin equation (overdamped Langevin will be easier and thus omitted):

Theorem H.1. *Consider, for $0 < \gamma < 2$, respectively SGHMC and EWSG,*

$$\begin{cases} \theta'_{k+1} &= \theta'_k + hr'_k \\ r'_{k+1} &= r'_k - h\gamma r'_k - h(\theta'_k - \mu_{I'_k}) + \sqrt{h}\sigma\xi'_{k+1} \end{cases}$$

and

$$\begin{cases} \theta_{k+1} &= \theta_k + hr_k \\ r_{k+1} &= r_k - h\gamma r_k - h(\theta_k - \mu_{I_k}) + \sqrt{h}\sigma\xi_{k+1} \end{cases},$$

where I'_k are i.i.d. uniform random variable on $[n]$, I_k are $[\theta, r]$ dependent random variable on $[n]$ satisfying $\mathbb{P}(I_k = i) = 1/n + \mathcal{O}(h^p)$, and ξ_{k+1}, ξ'_{k+1} are standard i.i.d. Gaussian random variables. Denote by $\bar{\theta'}_k = \mathbb{E}\theta'_k$, $\bar{r'}_k = \mathbb{E}r'_k$, $\bar{\theta}_k = \mathbb{E}\theta_k$, $\bar{r}_k = \mathbb{E}r_k$, $x'_k = [\bar{\theta'}_k, \bar{r'}_k]^T$, and $x_k = [\bar{\theta}_k, \bar{r}_k]^T$, then

$$x'_k = (I + Ah)^k x'_0, \quad \text{where } A = \begin{bmatrix} 0 & 1 \\ -1 & -\gamma \end{bmatrix}, \tag{14}$$

for small enough h, $\|x'_k\|$ converges to 0 exponentially with $k \to \infty$, and x_k converges at a comparable speed in the sense that $\|x_k - x'_k\| = \mathcal{O}(h^p)$ if $x_0 = x'_0$.

Proof. Taking the expectation of the $[\theta', r']$ iteration and using the fact that $\sum_i \mu_i = 0$ and hence $\mathbb{E}\mu_{I'_k} = 0$, one easily obtains (14). The geometric convergence of x'_k

thus follows from the fact that eigenvalues of $I + Ah$ have less than 1 modulus for small enough h.

Let $e_k = [0, \mathbb{E}\mu_{I_k}]^T$ and then

$$e_k = [0, \sum_{i=1}^{n} \mathbb{P}(I_k = i)\mu_i]^T = [0, \mathcal{O}(h^p)]^T$$

Now we take the expectation of both sides of the $[\theta, r]$ iteration and obtain $x_{k+1} = (I + Ah)x_k + he_k$. Therefore

$$\begin{aligned} x_k =& (I + Ah)^k x_0 + (I + Ah)^{k-1} h e_0 + \cdots + (I + Ah) h e_{k-2} + h e_{k-1} \\ =& x'_k + h\big((I + Ah)^{k-1} e_0 + \cdots + (I + Ah) e_{k-2} + e_{k-1}\big) \end{aligned}$$

To bound the difference, note $I + Ah$ is diagonalizable with complex eigenvalues $\lambda_{1,2}$ satisfying

$$|\lambda_1| = |\lambda_2| = \sqrt{1 - h\gamma + h^2} = 1 - \gamma h/2 + \mathcal{O}(h^2).$$

Projecting e_j to the corresponding eigenspaces via $e_j = v_{1,j} + v_{2,j}$, we can get

$$\begin{aligned} & h\|(I + Ah)^{k-1} e_0 + \cdots + e_{k-1}\| \\ & \leq h\left(\|(I + Ah)^{k-1} e_0\| + \cdots + \|e_{k-1}\|\right) \\ & = h\left(|\lambda_1|^{k-1}\|v_{1,0}\| + |\lambda_2|^{k-1}\|v_{2,0}\| + \cdots + \|v_{1,k-1}\| + \|v_{2,k-1}\|\right) \\ & \leq hCh^p(|\lambda_1|^{k-1} + \cdots + 1) = hCh^p \frac{1 - |\lambda_1|^k}{1 - |\lambda_1|} \leq hCh^p \frac{1}{1 - |\lambda_1|} \\ & \leq \hat{C}h^p \end{aligned}$$

for some constant C and $\hat{C}$. □

Important to note is, although this is already a nonlinear example for EWSG (as nonlinearity enters through the μ_{I_k} term), it is a linear example for SGHMC. For the fully nonlinear cases, a tight quantification of EWSG's convergence speed remains to be an open theoretical challenge (a loose quantification is already given by the general Theorem 4.3).

REFERENCES

[1] S. Ahn, A. Korattikara and M. Welling, Bayesian posterior sampling via stochastic gradient fisher scoring, In *29th International Conference on Machine Learning, ICML 2012*, (2012), 1591–1598.

[2] F. Bach, Stochastic gradient methods for machine learning, Technical report, INRIA - Ecole Normale Superieur, 2013. http://lear.inrialpes.fr/people/harchaoui/projects/gargantua/slides/bach_gargantua_nov2013.pdf.

[3] J. Baker, P. Fearnhead, E. Fox and C. Nemeth, Control variates for stochastic gradient MCMC, *Stat. Comput.*, **29** (2019), 599–615.

[4] R. Bardenet, A. Doucet and C. Holmes, On markov chain Monte Carlo methods for tall data, *J. Mach. Learn. Res.*, **18** (2017), 43 pp.

[5] V. S. Borkar and S. K. Mitter, A strong approximation theorem for stochastic recursive algorithms, *J. Optim. Theory Appl.*, **100** (1999), 499–513.

[6] N. Bou-Rabee, A. Eberle and R. Zimmer, Coupling and convergence for Hamiltonian Monte Carlo, *Ann. Appl. Probab.*, **30** (2018), 1209–1250.

[7] N. Bou-Rabee and H. Owhadi, Long-run accuracy of variational integrators in the stochastic context, *SIAM J. Numer. Anal.*, **48** (2010), 278–297.

[8] N. Bou-Rabee and J. M. Sanz-Serna, Geometric integrators and the Hamiltonian Monte Carlo method, *Acta Numer.*, **27** (2018), 113–206.

[9] S. Brooks, A. Gelman, G. Jones and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.
[10] N. Chatterji, N. Flammarion, Y. Ma, P. Bartlett and M. Jordan, On the theory of variance reduction for stochastic gradient monte carlo, *ICML*, 2018.
[11] C. Chen, N. Ding and L. Carin, On the convergence of stochastic gradient mcmc algorithms with high-order integrators, *Advances in Neural Information Processing Systems*, (2015), 2278–2286.
[12] T. Chen, E. B. Fox and C. Guestrin, Stochastic gradient hamiltonian monte carlo, *International Conference on Machine Learning*, (2014), 1683–1691.
[13] X. Cheng, N. S. Chatterji, Y. Abbasi-Yadkori, P. L. Bartlett and M. I. Jordan, Sharp convergence rates for langevin dynamics in the nonconvex setting, preprint, arXiv:1805.01648, 2018.
[14] X. Cheng, N. S. Chatterji, P. L. Bartlett and M. I. Jordan, Underdamped langevin mcmc: A non-asymptotic analysis, *Proceedings of the 31st Conference On Learning Theory, PMLR*, 2018.
[15] D. Csiba and P. Richtárik, Importance sampling for minibatches, *J. Mach. Learn. Res.*, **19** (2018), 21 pp.
[16] A. S. Dalalyan and A. Karagulyan, User-friendly guarantees for the langevin Monte Carlo with inaccurate gradient, *Stochastic Process. Appl.*, **129** (2019), 5278–5311.
[17] A. Defazio, F. Bach and S. Lacoste-Julien, Saga: A fast incremental gradient method with support for non-strongly convex composite objectives, In *Advances in Neural Information Processing Systems*, (2014), 1646–1654.
[18] A. Defazio and L. Bottou, On the ineffectiveness of variance reduced optimization for deep learning, In *Advances in Neural Information Processing Systems*, (2019), 1755–1765.
[19] K. A. Dubey, S. J. Reddi, S. A. Williamson, B. Poczos, A. J. Smola and E. P. Xing, Variance reduction in stochastic gradient langevin dynamics, In *Advances in Neural Information Processing Systems*, (2016), 1154–1162.
[20] T. Fu and Z. Zhang, Cpsg-mcmc: Clustering-based preprocessing method for stochastic gradient mcmc, In *Artificial Intelligence and Statistics*, (2017), 841–850.
[21] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato and Y. LeCun, What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, (2009), 2146–2153.
[22] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, *Advances in Neural Information Processing Systems*, (2013), 315–323.
[23] A. Korattikara, Y. Chen and M. Welling, Austerity in mcmc land: Cutting the metropolis-hastings budget, In *International Conference on Machine Learning*, (2014), 181–189.
[24] R. Kubo, The fluctuation-dissipation theorem, *Reports on Progress in Physics*, **29** (1966).
[25] C. Li, C. Chen, D. Carlson and L. Carin, Preconditioned stochastic gradient langevin dynamics for deep neural networks, In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[26] Q. Li, C. Tai and E. Weinan, Stochastic modified equations and adaptive stochastic gradient algorithms, In *J. Mach. Learn. Res.*, **20** (2019), 47 pp.
[27] R. Li, H. Zha and M. Tao, Mean-square analysis with an application to optimal dimension dependence of Langevin Monte Carlo, preprint, arXiv:2109.03839, 2021.
[28] M. Lichman et al, UCI Machine Learning Repository, 2013.
[29] Y.-A. Ma, T. Chen and E. B. Fox, A complete recipe for stochastic gradient mcmc, In *Advances in Neural Information Processing Systems*, (2015), 2917–2925.
[30] D. Maclaurin and R. P. Adams, Firefly monte carlo: Exact mcmc with subsets of data, In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
[31] S. Mandt, M. D. Hoffman and D. M. Blei, Stochastic gradient descent as approximate Bayesian inference, *J. Mach. Learn. Res.*, **18** (2017), 134, 35 pp.
[32] J. C. Mattingly, A. M. Stuart and M. V. Tretyakov, Convergence of numerical time-averaging and stationary measures via Poisson equations, *SIAM J. Numer. Anal.*, **48** (2010), 552–577.
[33] D. Needell, R. Ward and N. Srebro, Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm, *Math. Program.*, **155** (2016), 549–573.
[34] S. Patterson and Y. W. Teh, Stochastic gradient Riemannian Langevin dynamics on the probability simplex, *Advances in Neural Information Processing Systems*, (2013), 3102–3110.

[35] G. A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, Texts in Applied Mathematics, 60. Springer, New York, 2014.
[36] G. O Roberts, R. L. Tweedie, et al, Exponential convergence of Langevin distributions and their discrete approximations, *Bernoulli*, **2** (1996), 341–363.
[37] M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton and A. Sarkar, Non-uniform stochastic average gradient method for training conditional random fields, *Artificial Intelligence and Statistics*, (2015), 819–828.
[38] M. Schmidt, N. L. Roux and F. Bach, Minimizing finite sums with the stochastic average gradient, *Math. Program.*, **162** (2017), 83–112.
[39] M. Tao and T. Ohsawa, Variational optimization on lie groups, with examples of leading (generalized) eigenvalue problems, *AISTATS*, 2020.
[40] Y. W. Teh, A. H. Thiery and S. J. Vollmer, Consistency and fluctuations for stochastic gradient Langevin dynamics, *J. Mach. Learn. Res.*, **17** (2016), 7, 33 pp.
[41] 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** (2016), 159, 45 pp.
[42] M. Welling and Y. W. Teh, Bayesian learning via stochastic gradient langevin dynamics, *International Conference on Machine Learning*, (2001), 681–688.
[43] A. G. Wilson, The case for bayesian deep learning, preprint, arXiv:2001.10995, 2020.
[44] R. Zhang, A. F. Cooper and C. D. Sa, Asymptotically optimal exact minibatch metropolis-hastings, *NeurIPS*, 2020.
[45] R. Zhang and C. D. Sa, Poisson-minibatching for gibbs sampling with convergence rate guarantees, *NeurIPS*, 2019.
[46] P. Zhao and T. Zhang, Stochastic optimization with importance sampling for regularized loss minimization, *International Conference on Machine Learning*, (2015), 1–9.
[47] R. Zhu, Gradient-based sampling: An adaptive importance sampling for least-squares, In *Advances in Neural Information Processing Systems*, (2016), 406–414.

Received April 2021; revised September 2021; early access December 2021.

E-mail address: liruilin1993@gmail.com
E-mail address: xinwangmath@gmail.com
E-mail address: zhahy@cuhk.edu.cn
E-mail address: mtao@gatech.edu