Journal of Machine Learning Research 22 (2021) 1-51 Submitted 10/20; Revised 7/21; Published 9/21

Langevin Monte Carlo: random coordinate descent and variance reduction

Zhiyan Ding ZDING49@MATH.WISC.EDU
Mathematics Department
University of Wisconsin-Madison
Madison, WI 53705 USA.

Qin Li QINLI@MATH.WISC.EDU
Mathematics Department and Wisconsin Institutes for Discovery
University of Wisconsin-Madison
Madison, WI 53705 USA.

Editor: Anthony Lee

Abstract

Langevin Monte Carlo (LMC) is a popular Bayesian sampling method. For the log-concave distribution function, the method converges exponentially fast, up to a controllable discretization error. However, the method requires the evaluation of a full gradient in each iteration, and for a problem on $\mathbb{R}^d$, this amounts to d times partial derivative evaluations per iteration. The cost is high when $d \gg 1$. In this paper, we investigate how to enhance computational efficiency through the application of RCD (random coordinate descent) on LMC. There are two sides of the theory: By blindly applying RCD to LMC, one surrogates the full gradient by a randomly selected directional derivative per iteration. Although the cost is reduced per iteration, the total number of iteration is increased to achieve a preset error tolerance. Ultimately there is no computational gain; We then incorporate variance reduction techniques, such as SAGA (stochastic average gradient) and SVRG (stochastic variance reduced gradient), into RCD-LMC. It will be proved that the cost is reduced compared with the classical LMC, and in the underdamped case, convergence is achieved with the same number of iterations, while each iteration requires merely one directional derivative. This means we obtain the best possible computational cost in the underdamped-LMC framework.

Keywords: Langevin Monte Carlo, Random coordinate descent, Variance reduction, Bayesian inference, Wasserstein metric

1. Introduction

Monte Carlo Sampling is one of the core problems in Bayesian statistics, data assimilation (Reich, 2011), and machine learning (Andrieu et al., 2003), with wide applications in atmospheric science (Fabian, 1981), petroleum engineering (Nagarajan et al., 2007), epidemiology (Li et al., 2020), in the form of inverse problems, volume computation (Vempala, 2010), and bandit optimization (Mazumdar et al., 2020; Russo et al., 2018; May et al., 2012).

Let $f(x)$ be μ-strongly convex with its gradient being L-Lipschitz in $\mathbb{R}^d$, and define the target density function $p \propto e^{-f}$. Then $p(x)$ is a log-concave probability density function. To sample from the distribution induced by $p(x)$ amounts to finding an $x \in \mathbb{R}^d$ (or a list of samples that can be regarded as i.i.d. (independent and identically distributed) drawn from this distribution.

The literature is very rich on sampling, and there are many approaches (Neal, 2001, 1993; Roberts and Rosenthal, 2004; Metropolis et al., 1953; Hastings, 1970; Geman and Geman, 1984; Duane et al., 1987; Neal, 2012). In this paper, we only study variations of Langevin Monte Carlo (LMC) (Rossky et al., 1978; Parisi, 1981; Roberts and Tweedie, 1996b; Chen et al., 2014; Ma et al., 2015). LMC, including the classical LMC and its variations, can be regarded as one subcategory of Markov chain Monte Carlo (MCMC). They are attractive mostly due to the fast convergence rate. Indeed, the classical LMC and its most variations can be regarded as discrete versions of some Langevin dynamics, a set of stochastic differential equations (SDE) that roughly follow the gradient of f, with some added Brownian motion terms. Usually, the SDEs are designed so the distribution converges exponentially fast in time to the target distribution (Markowich and Villani, 1999). LMC, therefore, viewed as the discrete versions (such as the Euler-Maruyama method) of the SDEs, converge also almost exponentially fast, up to a discretization error. The non-asymptotic analysis for the methods is studied in detail in recent years (Dalalyan and Riou-Durand, 2020; Dalalyan and Karagulyan, 2019; Durmus et al., 2019; Cheng et al., 2018; Dalalyan and Riou-Durand, 2020; Eberle et al., 2019; Dwivedi et al., 2018).

However, like many other sampling methods, the numerical cost of classical (both overdamped and underdamped) LMC methods increases with respect to d, the dimensionality of the problem. While the number of iterations to achieve a preset error tolerance already depends on d, the cost per iteration may also increase with respect to d. Indeed, there are examples in which, to compute the gradients, d partial derivatives need to be computed separately (see Section 1.2), and this adds another d folds of cost per iteration.

The cost of evaluating the gradient has triggered a large number of studies centering around the “gradient-free" property. Many approaches were investigated, including Important Sampling (Geweke, 1989; Neal, 2001; Moral et al., 2006; Murray et al., 2016), ensemble methods (Iglesias et al., 2013; Garbuno-Inigo et al., 2020; Reich, 2011), and random walks (Mengersen and Tweedie, 1996; Roberts and Tweedie, 1996b,a; Dwivedi et al., 2018). These methods shed light on eliminating the evaluation of the gradients, but to this day we have not found a method that is proved to be more competitive with LMC.

1.1 Contribution

In this article, we would like to study how to enhance the numerical efficiency of LMC, with a special eye on improving the cost's dependence on d. The strategy we take is to develop alternatives for evaluating the gradients. More specifically, we explore how to incorporate random coordinate descent (RCD) in LMC. RCD is a technique developed for optimization problems, and it randomly chooses one partial derivative, as a replacement of the full gradient in gradient descent (GD). Since only one partial derivative is calculated in each iteration, the cost is reduced by d-folds per iteration. If one carefully controls the iteration number, the overall cost can also be reduced in certain scenarios, as proved in the optimization literature (Nesterov, 2012).

We study if applying RCD to LMC can bring us similar benefits. The theoretical guarantee we obtain in this paper shows that there are two sides of the story.

Firstly, we blindly apply RCD to LMC. This is to replace the full gradient in LMC with a randomly selected partial derivative in each iteration, exactly as RCD in optimization. This is done to both the overdamped LMC and underdamped LMC, resulting in two algorithms: RCD-O-LMC, and RCD-U-LMC respectively. Since only one directional derivative is calculated instead of the full gradient in each iteration, the cost is reduced by d-folds per iteration. However, we can find counterexamples that show this blind application of RCD induces a large error term that represents the high variance produced by the random direction selection procedure. Exactly due to this large error term, more iterations are required. Ultimately there is no saving in the numerical cost.

Secondly, we study variance reduction techniques. Two main techniques are explored: SAGA (stochastic average gradient) (Schmidt et al., 2017; Defazio et al., 2014) and SVRG (stochastic variance reduced gradient) (Johnson and Zhang, 2013), hoping to reduce the large error term mentioned above, saving the numerical cost in the end. These methods were developed as optimization techniques to improve the convergence rate of SGD (stochastic gradient descent) (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952; Bottou, 2010), an algorithm that looks for the minimizer of an objective function that has the form of $f(x) = \sum_{n=1}^{N} f_n(x)$. Such techniques were then applied to improve stochastic gradient Markov chain Monte Carlo (SG-MCMC) (Welling and Teh, 2011; Dubey et al., 2016; Durmus et al., 2016; Baker et al., 2019). The complexities of these methods are studied in (Chatterji et al., 2018; Zou et al., 2018a,b; Zou and Gu, 2021). A summary of these results can be found in (Zou and Gu, 2021, Section 2.3). We investigate how to incorporate these techniques to enhance the performance of RCD-LMC. In our case f does not necessarily have the form of $\sum_{n}^{N} f_n(x)$, however, viewing $\nabla f = \sum_k \partial_k f \mathbf{e}^k$, variance reduction techniques for SGD can still be borrowed. The methods with variance reduction integrated are called Randomized Coordinates Averaging Descent Overdamped/Underdamped LMC (RCAD-O/U-LMC) and Stochastic Variance Reduced Gradient Overdamped/Underdamped LMC (SVRG-O/U-LMC). We will show that, with either variance reduction technique, in the underdamped setting, the new methods require the same number of iterations as the classical U-LMC (Cheng et al., 2018) to achieve a small preset error tolerance ϵ, but the number of directional derivative per iteration is 1 instead of d. This automatically saves d folds of computation.

We summarize all the convergence results in Table 1 (assuming computing the full gradient costs d times of computing one directional derivative and arbitrary initial distribution). We do not present the dependence on the conditioning κ in the table, but they are included in the later discussions. The presented result assumes the large κ is a secondary concern compared to the large d ($\kappa \ll d$).

1.2 Discussions on assumptions, and relation to the literature

Throughout the paper we assume that computing one partial derivative costs $1/d$ of computing the full gradient. This assumption may not hold in some applications in which the full gradient can be computed efficiently. This happens in logistic regression, support vector machines (SVM), and deep learning where backward propagation is heavily relied on (Rumelhart et al., 1986).

However, there are many examples where this assumption indeed holds true. For example, given a graph with nodes $\mathcal{N} = \{1, 2, \ldots, d\}$ and directed edges $\mathcal{E} \subset \{(i,j) : i, j \in \mathcal{N}\}$,

Algorithm	Number of iterations	Cost
O-LMC	$\widetilde{O}\left(d/\epsilon\right)$	$\widetilde{O}\left(d^2/\epsilon\right)$
U-LMC	$\widetilde{O}\left(d^{1/2}/\epsilon\right)$	$\widetilde{O}\left(d^{3/2}/\epsilon\right)$
RCD-O-LMC	$\widetilde{O}\left(d^2/\epsilon^2\right)$	$\widetilde{O}\left(d^2/\epsilon^2\right)$
RCD-U-LMC	$\widetilde{O}\left(d^2/\epsilon^2\right)$	$\widetilde{O}\left(d^2/\epsilon^2\right)$
SVRG-O-LMC	$\widetilde{O}\left(d^{3/2}/\epsilon\right)$	$\widetilde{O}\left(d^{3/2}/\epsilon\right)$
SVRG-U-LMC	$\widetilde{O}\left(\max\{d^{4/3}/\epsilon^{2/3}, d^{1/2}/\epsilon\}\right)$	$\widetilde{O}\left(\max\{d^{4/3}/\epsilon^{2/3}, d^{1/2}/\epsilon\}\right)$
RCAD-O-LMC	$\widetilde{O}\left(d^{3/2}/\epsilon\right)$	$\widetilde{O}\left(d^{3/2}/\epsilon\right)$
RCAD-U-LMC	$\widetilde{O}\left(\max\{d^{4/3}/\epsilon^{2/3}, d^{1/2}/\epsilon\}\right)$	$\widetilde{O}\left(\max\{d^{4/3}/\epsilon^{2/3}, d^{1/2}/\epsilon\}\right)$

Table 1: Number of iterations and numerical cost to achieve ϵ-accuracy. The results for the classical O-LMC and U-LMC come from (Dalalyan and Karagulyan, 2019) and (Cheng et al., 2018; Dalalyan and Riou-Durand, 2020) respectively. The notation $\widetilde{O}(f)$ omits the possible log terms. For the overdamped cases, we assume the Lipschitz continuity for both the gradient term and the hessian term, and for the underdamped cases, Lipschitz continuity is only assumed for the gradient term.

suppose there is a scalar variable x_i associated with each node $i = 1, 2, \ldots, d$, and that the function f has the form

$$f(x) = \sum_{(i,j)\in\mathcal{E}} f_{ij}(x_i, x_j)\,.$$

Then the partial derivative of f with respect to x_i is given by

$$\frac{\partial f}{\partial x_i} = \sum_{j:(i,j)\in\mathcal{E}} \frac{\partial f_{ij}}{\partial x_i}(x_i, x_j) + \sum_{l:(l,i)\in\mathcal{E}} \frac{\partial f_{li}}{\partial x_i}(x_l, x_i)\,.$$

Note that the number of terms in the summations in this expression equals the number of edges in the graph that touch node i, the expected value of which is about $2/d$ times the total number of edges in the graph. Meanwhile, evaluation of the full gradient would require evaluation of partial derivatives with respect to both component i and j of the function f_{ij} for *all* edges in the graph, leading to a factor-of-d difference in evaluation cost. This setup is encountered in many graph-based problems, such as graph-based label propagation in semi-supervised learning (Chapelle et al., 2006), finding the densest k-subgraph (Sotirov, 2020), and finding the most likely assignments in continuous pairwise graphical models (Rue and Held, 2005). It is also observed in large-scale SVM problems (when solved in dual form) (Hsieh et al., 2008) as well.

The assumption also incurs in many PDE-constrained inverse problems as well. The evaluation of the objective function f typically accounts for one PDE solve. As a comparison, each partial derivative amounts to 2 PDE solves (one forward encoding the input data and one adjoint presenting the output test function), and in each iteration, to have the full gradient (also termed Fréchet derivative in that context), many partial derivatives need to be computed (Leugering et al., 2012). This leads to a large number of PDE solves for only one iteration.

Another assumption we make in this paper is that the conditioning of the objective function f is only moderately big, in the sense that its value is not comparable to the largeness of d. This means the underperformance of RCD-LMC and the outperformance of the variance reduced RCD-LMC hold uniformly true for all f that have moderate condition numbers. If f has bad conditioning, the Lipschitz constants for each direction are then drastically different. When this happens, one could potentially choose coordinates at different rates to reflect the skewness of f. This indeed happens in RCD surrogating GD, where the stiffer directions get chosen more frequently. If we drop the assumption and let f be very skewed as well, a biased selection process could potentially enhance our methods even more. This topic is discussed in two separate contributions, see (Ding et al., 2021a) and (Ding et al., 2021b) for overdamped and underdamped cases respectively. For the completeness of the paper, in Remark 5 we discuss how to sample from a non-uniform coordinate selection process, and in Remark 19 we argue why it does not bring computational benefits when f is well-conditioned.

There are strong connections between the methods proposed in the paper and the existing literature. We first mention that most of the techniques are borrowed from optimization. RCD is one strategy that surrogates the full gradient by a partial derivative in GD and has been shown to be useful when the objective function f has certain structures. Similar ideas were used in SGD when the objective function has the form of $f = \sum_{k=1}^{K} f_k$, and in SGD, a ∇f_i is randomly selected to surrogate ∇f per iteration. It was observed that although SGD reduces the per iteration cost by K, the total number of iteration is increased due to the induced higher variance. For this reason, multiple variance reduction techniques were developed, including SVRG (stochastic variance reduced gradient) (Johnson and Zhang, 2013), and SAGA (stochastic average gradient) (Schmidt et al., 2017; Defazio et al., 2014), all adopted in the current paper.

We also would like to mention that the authors of the paper have already presented part of the results of this paper in (Ding and Li, 2020). In that paper, it was already observed that blindly applying RCD to LMC will not enhance the efficiency, and some variance reduction is needed. RCAD-O/U-LMC, built upon integrating SAGA to RCD-LMC, was proposed and studied there. That work was our first attempt in addressing similar issues. With the insights gained there, we greatly extended the results, and provide the general recipe in the current paper. The main extension can be summarized as follows:

- We give the non-asymptotic convergence result of RCD-O/U-LMC in Theorem 3 and Theorem 6. These two bounds were not included in (Ding and Li, 2020).

- We propose and rigorously analyze a new variance reduction method SVRG-O/U-LMC. This method, similar to RCAD-O/U-LMC, can also be shown to reduce the computational cost.

- Moreover, we present a systematic understanding of how the variance presents in the convergence rate, and how to integrate variance reduction techniques to reduce the cost. The overarching results that summarize, in a general form, the impact of involving randomness in the coordinate selection process are presented in Theorem 15 and Theorem 23. The two theorems address the overdamped and the underdamped cases respectively. The blind application of RCD, and the incorporation of SAGA and SVRG, then can be viewed as three different examples under this framework. We emphasize that though we only study two variance reduction techniques in the current paper, Theorem 15 and 23 are general enough to treat other methods as well. Any new variance reduction method, when cast into the random coordinate LMC framework, can be analyzed in a similar fashion.

1.3 Organization

In Section 2, we discuss the essential ingredients of our methods. In Section 3, we unify the notations and assumptions used in our methods. In Section 4, we present the convergence results of RCD-LMC and also provide a counter-example to demonstrate the fact that RCD-LMC does not save numerical cost. In Section 5, we introduce the methods that incorporate variance reduction techniques: SVRG-O/U-LMC and RCAD-O/U-LMC, and present the theorems on the convergence and the numerical cost. In Section 6, numerical evidence will be given to demonstrate the improved efficiency of our new methods. In Section 7-8, we provide the proof of the theorems for overdamped, and underdamped settings respectively. We note that in subsection 7.1 and subsection 8.1 we provide the overarching results for O/U-LMC when randomness is involved in the coordinate selection process. The later subsections are devoted to the realization of these two theorems on specific algorithms as examples. The discussion on the best possible numerical cost is heavily technical and thus delayed to these two subsections (Remark 16 and Remark 24). Some technical lemmas used in these two sections are postponed to the appendix.

2. Essential ingredients

In this section, we discuss the essential ingredients of our methods: the overdamped and underdamped Langevin dynamics and the associated Monte Carlo methods (O-LMC and U-LMC); random coordinate descent (RCD); and variance reduction techniques, including both SVRG and SAGA.

2.1 Overdamped Langevin dynamics and O-LMC

The O-LMC method comes from the overdamped Langevin dynamics, a stochastic differential equation that writes:

$$\mathrm{d}X_t = -\nabla f(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,. \tag{1}$$

This SDE characterizes the trajectory of X_t. Two forcing terms $\nabla f(X_t)\,\mathrm{d}t$ and $\mathrm{d}B_t$ compete: the former drives X_t to the minimum of f and the latter provides Brownian motion and thus small oscillations along the trajectory. The initial data X_0 is a random variable drawn from a given distribution induced by $q_0(x)$. Denote $q(x,t)$ the probability density function of X_t, it is a well-known result that $q(x,t)$ satisfies the following Fokker-Planck equation:

$$\partial_t q = \nabla\cdot(\nabla f q + \nabla q)\,, \quad \text{with} \quad q(x,0) = q_0\,, \tag{2}$$

and furthermore, $q(x,t)$ converges to the target density function $p(x)\propto e^{-f}$ exponentially fast in time (Markowich and Villani, 1999).

The overdamped Langevin Monte Carlo (O-LMC), as a sampling method, is simply a discrete-in-time version of the SDE (1). A standard Euler-Maruyama method applied on the equation gives:

$$x^{m+1} = x^m - \nabla f(x^m)h + \sqrt{2h}\xi^m\,, \tag{3}$$

where h is the small time-step and ξ^m is i.i.d. drawn from $\mathcal{N}(0, I_d)$ with I_d being the identity matrix of size d. Since (3) approximates (1), the density of x^m becomes $p(x)$ as $m \to \infty$, up to a discretization error. It was proved in (Dalalyan and Karagulyan, 2019) that the convergence to ϵ is achieved within $\widetilde{O}(d/\epsilon)$ iterations if the hessian of f is Lipschitz. If this hessian is not Lipschitz, the number of iterations is shown to be $\widetilde{O}(d/\epsilon^2)$. In many real applications, the gradient of f is not explicitly known and some approximation is used, introducing another layer of numerical error. In (Dalalyan and Karagulyan, 2019), the authors discussed the effect of such error, under the assumption that the error term has a bounded variance.

2.2 Underdamped Langevin dynamics and U-LMC

The underdamped Langevin dynamics (Chen et al., 2014) is characterized by the following SDE system:

$$\begin{cases} \mathrm{d}X_t = V_t\,\mathrm{d}t \\ \mathrm{d}V_t = -2V_t\,\mathrm{d}t - \gamma\nabla f(X_t)\,\mathrm{d}t + \sqrt{4\gamma}\,\mathrm{d}B_t \end{cases}, \tag{4}$$

where $\gamma > 0$ is a parameter to be tuned. Denote $q(x, v, t)$ the probability density function of (X_t, V_t), then q satisfies the Fokker-Planck equation

$$\partial_t q = \nabla \cdot \left(\begin{bmatrix} -v \\ 2v + \gamma\nabla f \end{bmatrix} q + \begin{bmatrix} 0 & 0 \\ 0 & 2\gamma \end{bmatrix} \nabla q \right)\,,$$

and under mild conditions, in $t \to \infty$, it converges to $p_2(x, v) \propto \exp(-(f(x) + |v|^2/2\gamma))$, making the marginal density function for x the target $p(x)$.

The underdamped Langevin Monte Carlo algorithm, U-LMC, can be viewed as a numerical solver to (4). In each step, we sample $(x^{m+1}, v^{m+1}) \in \mathbb{R}^{2d}$ as a Gaussian random variable determined by (x^m, v^m) with the following expectation and covariance:

$$\begin{aligned} &\mathbb{E}x^{m+1} = x^m + \frac{1}{2}\left(1 - e^{-2h}\right)v^m - \frac{\gamma}{2}\left(h - \frac{1}{2}\left(1 - e^{-2h}\right)\right)\nabla f(x^m)\,, \\ &\mathbb{E}v^{m+1} = v^m e^{-2h} - \frac{\gamma}{2}\left(1 - e^{-2h}\right)\nabla f(x^m)\,, \\ &\mathrm{Cov}\left(x^{m+1}\right) = \gamma\left[h - \frac{3}{4} - \frac{1}{4}e^{-4h} + e^{-2h}\right]\cdot I_d\,,\ \mathrm{Cov}\left(v^{m+1}\right) = \gamma\left[1 - e^{-4h}\right]\cdot I_d\,, \\ &\mathrm{Cov}\left(x^{m+1}, v^{m+1}\right) = \frac{\gamma}{2}\left[1 + e^{-4h} - 2e^{-2h}\right]\cdot I_d\,. \end{aligned} \tag{5}$$

We here used the notation $\mathbb{E}$ to denote the expectation, and $\mathrm{Cov}(a, b)$ to denote the covariance of a and b. If $b = a$, we abbreviate it to $\mathrm{Cov}(a)$. The scheme can be interpreted as sampling from the following dynamics in each time interval:

$$\begin{cases} \mathrm{X}_t = x^m + \int_0^t \mathrm{V}_s \,\mathrm{d}s \\ \mathrm{V}_t = v^m e^{-2t} - \frac{\gamma}{2}(1-e^{-2t})\nabla f(x^m) + \sqrt{4\gamma} e^{-2t} \int_0^t e^{2s}\,\mathrm{d}B_s \end{cases}$$

with $x^{m+1} = \mathrm{X}_h$ and $v^{m+1} = \mathrm{V}_h$, and h is the time step.

The underdamped Langevin Monte Carlo demonstrates a faster convergence rate (Cheng et al., 2018) than O-LMC. Due to the introduction of V in (4), the trajectory X_t is smoother than that of V_t whose smoothness is determined by the Brownian motion B_t. As a result, a higher-order discretization is possible for this augmented dynamics. Without the assumption on the hessian of f being Lipschitz, the number of iterations is $\widetilde{O}(\sqrt{d}/\epsilon)$ to achieve ϵ accuracy. A high (3rd) order discretization was discussed for (4) in (Mou et al., 2021), further enhancing the numerical accuracy to $\widetilde{O}(d^{1/4}/\epsilon^{1/2})$ when f is smooth enough. Similar to the discussion for O-LMC in (Dalalyan and Karagulyan, 2019), the authors in (Dalalyan and Riou-Durand, 2020) also studied the error in estimating $\nabla f(x^m)$, but they also assumed the variance is bounded.

2.3 Random coordinate descent (RCD)

The idea of RCD is to surrogate the full gradient in Gradient Descent by a randomly selected partial derivative in each iteration (Nesterov, 2012). Per iteration, only one partial derivative is computed instead of d, so there is some hope that total cost can be reduced. More specifically one approximates

$$\nabla f \approx d\left(\nabla f(x) \cdot \mathbf{e}^r\right)\mathbf{e}^r\,, \tag{6}$$

where $\mathbf{e}^i$ is the i-th unit direction and r is randomly drawn from $1, 2, \cdots, d$. This approximation is consistent in the expectation sense because

$$\mathbb{E}_r\left(d\left(\nabla f(x)\cdot \mathbf{e}^r\right)\mathbf{e}^r\right) = \nabla f(x)\,.$$

It was shown in (Wright, 2015; Richtárik and Takáč, 2011; Nesterov, 2012) that cost reduction is indeed observed, especially when the objective function f is highly skewed in a high dimensional space. Similar ideas were discussed in (Gerencsér, 1997; Kleinman et al., 1999) where SPSA, simultaneous perturbation stochastic approximation, was proposed.

There are many different ways to compute directional derivatives. Automatic differentiation is a strategy used often when the underlying function is composed of many simpler operations, and such composition is explicitly known (Baydin et al., 2018). When the function is complicated itself, one could use the most basic finite differencing method. That is to use $\partial_i f(x) \approx \frac{f(x+\eta\mathbf{e}^i)-f(x-\eta\mathbf{e}^i)}{2\eta}$ at a sacrifice of $O(\eta^2)$ numerical error (assuming enough smoothness of f). For special problems such as the PDE-based inverse problems, raising in atmospheric science and epidemiology (Li et al., 2020), one could further translate the derivative computation into a combination of one forward and one adjoint PDE solves (Martin et al., 2012). Whichever strategy one employs, in some applications, it is true that when ∇f is not explicitly known, the computation of the full gradient costs d times of computing one directional derivative. This makes the saving on d extremely important.

2.4 Variance reduction techniques: SVRG and SAGA

SVRG (stochastic variance reduced gradient)(Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014) (a modified version of SAG (Schmidt et al., 2017), stochastic average gradient) are optimization techniques that are widely used in reducing variance in randomized solvers. In particular, they are introduced to enhance the numerical performance of SGD (stochastic gradient descent). SGD is a stochastic version of Gradient Descent (GD) that looks for the minimization of an objective function that has the form of $f = \sum_{n=1}^N f_n$ with $N \gg 1$. GD requires $\nabla f = \sum_{n=1} \nabla f_n$ evaluated at the current sample in each iteration, which amounts to a computation of N gradients. In SGD, one merely randomly selects one ∇f_r to represent the full ∇f. Per iteration, this reduces the cost by N folds. However, due to the random selection process, high variance is induced and that brings large errors. More iterations are then needed to achieve a preset accuracy. SVRG and SAGA are algorithms proposed to reduce this variance, hoping to eliminate the error induced by the randomness and save computational cost overall. We now describe the procedure of SVRG and SAGA.

In SVRG, one has a preset iteration number τ, and the full gradient ∇f is computed only once every τ steps. Between the epochs, per step, only one f_r is selected at random to represent the new gradient. In particular, call $\widetilde{x}$ the sample obtained at $k\tau$-th step with k being an integer, we have the full gradient at this point $\nabla f(\widetilde{x})$. In the following $\tau - 1$ steps, per iteration, one f_r is chosen at random and the new gradient is approximated by

$$\nabla f(x) \approx \nabla f(\widetilde{x}) + N\left(\nabla f_r(x) - \nabla f_r(\widetilde{x})\right) . \tag{7}$$

After τ steps, $\widetilde{x}$ is updated and one evaluates the full $\nabla f(\widetilde{x})$ again. This approximation (7) is unbiased if r is chosen uniformly in the sense that

$$\nabla f(x) = \sum_n \nabla f_n(x) = \mathbb{E}_r\left[\nabla f(\widetilde{x}) + N\left(\nabla f_r(x) - \nabla f_r(\widetilde{x})\right)\right] .$$

SAGA is slightly different: it only requires the computation of the full gradient at the initial step. In the later iterations, per iteration, only one f_r is chosen at random to update the full gradient while others are kept untouched. Term $\{g^n_m\}_{i=1}^N$ the m-th step approximation to ∇f_n, then $g^n_0 = \nabla f_n(x^0)$. For the following iterations with $m > 1$, r is uniformly randomly picked and $g^r_m = \nabla f_r(x^m)$ while others are kept untouched $g^k_m = g^k_{m-1}$. We then approximate the full gradient using:

$$\nabla f(x^m) \approx \sum_{n=1}^N g^n_{m-1} + N(g^r_m - g^r_{m-1}) . \tag{8}$$

Similar to SVRG, the approximation (8) is unbiased in the sense that

$$\nabla f(x) = \mathbb{E}_r\left[\sum_{n=1}^N g^n_{m-1} + N\left(\nabla f_r(x) - g^r_{m-1}\right)\right] .$$

Both techniques are proven to reduce variance for SGD (Johnson and Zhang, 2013; Defazio et al., 2014).

3. Notations and classical results

We unify notions and assumptions, and briefly summarize the classical results in this section.

3.1 Assumptions and the Wasserstein distance

We make standard assumptions on $f(x)$:

Assumption 3.1 *The function f is second-order differentiable, μ-strongly convex and has an L-Lipschitz gradient:*

- *Convex, meaning for any $x, x' \in \mathbb{R}^d$:*

$$f(x) - f(x') - \nabla f(x')^\top (x - x') \geq (\mu/2)|x - x'|^2 \,. \tag{9}$$

- *Gradient is Lipschitz, meaning for any $x, x' \in \mathbb{R}^d$:*

$$|\nabla f(x) - \nabla f(x')| \leq L|x - x'| \,. \tag{10}$$

These assumptions together mean $\mu I_d \preceq H(f) \preceq L I_d$ where $H(f)$ is the hessian of f. We also define condition number of $f(x)$ as

$$\kappa = L/\mu \geq 1 \,. \tag{11}$$

Furthermore, for O-LMC we assume Lipschitz condition of the hessian:

Assumption 3.2 *Hessian of f is H-Lipschitz, meaning for any $x, x' \in \mathbb{R}^d$:*

$$\|H(f)(x) - H(f)(x')\|_2 \leq H|x - x'| \,. \tag{12}$$

Throughout the analysis, we use the Wasserstein distance as the quantity to measure the distance between two probability measures. For $p \geq 1$, define the Wasserstein distance to be:

$$W_p(\mu, \nu) = \left(\inf_{(X,Y) \in \Gamma(\mu,\nu)} \mathbb{E}|X - Y|^p \right)^{1/p} \,,$$

where $\Gamma(\mu, \nu)$ is the set of distribution of $(X, Y) \in \mathbb{R}^{2d}$ whose marginal distributions are μ and ν respectively for X and Y. These distributions are called the couplings of μ and ν. Here μ and ν can be either probability measures themselves or the measures induced by probability density functions μ and ν. In this paper, we only use $p = 2$.

3.2 Classical O/U-LMC

We now review the classical results regarding O-LMC and U-LMC. The two algorithm are summarized in the following:

The non-asymptotic convergence results have been thoroughly discussed in (Dalalyan and Karagulyan, 2019; Durmus et al., 2019; Durmus and Moulines, 2019) and (Cheng et al., 2018; Dalalyan and Riou-Durand, 2020) for O-LMC and U-LMC respectively. We merely cite the theorems.

Algorithm 1 Overdamped/Underdamped Langevin Monte Carlo (O/U-LMC)

```
Preparation:
1. Input: $h$ (time stepsize); $\gamma$ (parameter); $d$ (dimension); $M$ (stopping index); $\nabla f(x)$.
2. Initial: (overdamped): $x^0$ i.i.d. sampled from an initial distribution induced by $q_0(x)$.
        (underdamped): $(x^0, v^0)$ i.i.d. sampled from the initial distribution induced by $q_0(x, v)$.
Run: For $m = 0, 1, \cdots, M$
        (overdamped): Draw $\xi^m$ from $\mathcal{N}(0, I_d)$:
            $x^{m+1} = x^m - \nabla f(x^m)h + \sqrt{2h}\xi^m\,.$    (13)
    (underdamped): Sample $(x^{m+1}, v^{m+1})$ as Gaussian random variables with expectation and
    covariance defined in (5).
end
Output: $\{x^m\}$.
```

Theorem 1 *[(Dalalyan and Karagulyan, 2019) Theorem 5] Assume $h < \frac{2}{\mu+L}$ and f satisfies Assumptions 3.1-3.2. Denote $q^O_m(x)$ the probability density function of x^m computed using O-LMC, and define $W_m = W_2(q^O_m, p)$, the L_2-Wasserstein distance between $q^O_m(x)$ and p, then we have:*

$$W_m \leq \exp\left(-\mu m h\right) W_0 + \left[\frac{Hhd}{2\mu} + 3\kappa^{3/2}\mu^{1/2}hd^{1/2}\right]\,. \tag{14}$$

Note that there are two parts in the iteration formula for W_m, the exponentially decaying term, and the remainder term. Without considering the conditioning constants, the remainder term is of the order of hd. To have the Wasserstein distance to be smaller than a preset tolerance, for example, $W_m \leq \epsilon$, we need

$$\exp\left(-\mu m h\right) \lesssim \epsilon\,, \quad \text{and} \quad hd \lesssim \epsilon\,,$$

which explains $m = \widetilde{O}(d/\epsilon)$ in Table 1. Since each iteration requires the computation of the full gradient, meaning d partial derivatives, the total cost is then $\widetilde{O}(d^2/\epsilon)$. In Durmus et al. (2019), the authors improved the dependence of the convergence rate on the condition number, with the Lipschitz continuity of the hessian term relaxed.

For the underdamped LMC, we have:

Theorem 2 *[(Dalalyan and Riou-Durand, 2020) Theorem 2] Assume f satisfies Assumption 3.1, $\gamma = \frac{1}{L}$ and $h \leq \frac{1}{8\kappa^2\mu}$. Denote $q^U_m(x, v)$ the probability density function of (x^m, v^m) computed using U-LMC, and define $W_m = W_2(q^U_m, p)$, then we have:*

$$W_m \leq \sqrt{2}\exp(-0.375mh/\kappa)W_0 + h(\kappa d)^{1/2}\,. \tag{15}$$

The iteration formula is once again composed of two terms: the exponential term and the remainder, with the remainder being of order $hd^{1/2}$. Setting both terms smaller than ϵ, we have

$$h = \widetilde{O}(d^{-1/2}\epsilon)\,, \quad \text{and} \quad m = \widetilde{O}(d^{1/2}/\epsilon)\,,$$

making the total number of partial derivative calculations being $\widetilde{O}(d^{3/2}/\epsilon)$, as shown in Table 1.

4. Algorithms and Results of RCD-LMC

As seen in Algorithm 1, per iteration, ∇f needs to be evaluated at the current sample, and that prompts d partial derivative calculations. If d is big, the computation is expensive.

RCD (random coordinate descent) is a method introduced in optimization to reduce the number of partial derivative calculations from the classical gradient descent. It essentially surrogates the full gradient in the gradient descent method by one directional derivative in each iteration, and thus it naturally reduces the computational cost by d folds per iteration. If we blindly apply this approach to O/U-LMC, we arrive at RCD-O/U-LMC. These two methods were briefly discussed in (Ding and Li, 2020). It was also shown there that the methods, despite easy to implement and reduce the per-iteration-cost by d folds, the overall cost is, however, not saved due to the larger number of required iterations. This means blindly applying RCD to O/U-LMC is a bad strategy. For the completeness of the paper, we still summarize the results and cite the convergence rate in the following subsections.

We note that the increased numerical cost could be explained by the high variance induced by the randomness in the coordinate selection process. The same issue was encountered in optimization as well, see discussions in (Richtárik and Takáč, 2011; Wright, 2015). As a consequence, a lot of efforts have been placed on tuning the probability of the coordinate-drawing, such as incorporating the information hidden in the directional Lipschitz constants. The application of these strategies should be explored in sampling as well, but it is beyond the scope of the current paper. We should also emphasize that the counterexample is a worst-case study. There are plenty of examples where the application of RCD to sampling already gives a numerical boost, but in this paper, we would like to be as general as possible.

4.1 Algorithm

We apply RCD to both O-LMC and U-LMC. This amounts to replacing the gradient terms in (3) and (5) using the approximation (6) . The new methods are termed RCD-O-LMC and RCD-U-LMC respectively, and we present them in Algorithm 2.

4.2 Convergence results for RCD-O/U-LMC

We discuss the convergence of Algorithm 2 in this section and compare the results with the classical results for O-LMC (Dalalyan and Karagulyan, 2019) and U-LMC (Cheng et al., 2018).

4.2.1 Convergence for RCD-O-LMC

The theorem we show is:

Theorem 3 *Suppose f satisfies Assumptions 3.1-3.2, and h satisfies*

$$h < \min\left\{\frac{1}{9\kappa^2\mu d}, \frac{2}{H^2/(\kappa\mu^2)+\kappa^2\mu/d}\right\}\,, \tag{18}$$

then we have:

$$W_m \leq \exp\left(-\mu m h/4\right) W_0 + 6d(\kappa h)^{1/2}\,, \tag{19}$$

where $W_m = W_2(q^O_m, p)$ and $q^O_m(x)$ is the probability density function of x^m computed by RCD-O-LMC.

Algorithm 2 RCD-O/U-LMC

```
Preparation:
1. Input: $h$ (time stepsize); $\gamma$ (parameter); $d$ (dimension); $M$ (stopping index); $f(x)$.
2. Initial: (overdamped): $x^0$ i.i.d. sampled from an initial distribution induced by $q_0(x)$.
      (underdamped): $(x^0, v^0)$ i.i.d. sampled from the initial distribution induced by $q_0(x,v)$.
Run: For $m = 0\,,1\,,\cdots\,,M$
    1. Prepare directional derivative: draw $r$ uniformly from $1\,,\cdots\,,d$, and compute:
           $$F^m = d\partial_r f(x^m)\mathbf{e}^r\,. \qquad (16)$$
    2. (overdamped): Draw $\xi^m$ from $\mathcal{N}(0, I_d)$:
           $$x^{m+1} = x^m - F^m h + \sqrt{2h}\xi^m\,. \qquad (17)$$
    (underdamped): Sample $(x^{m+1}, v^{m+1})$ as Gaussian random variables with expectation and
    covariance defined in (5), replacing $\nabla f(x^m)$ by $F^m$.
end
Output: $\{x^m\}$.
```

We discuss the proof in Section 7.2. The statement serves as the guidance to tune parameters and estimates the computational complexity. According to (19), for ϵ accuracy, it suffices to choose

$$6\kappa d h^{1/2} \leq \frac{\epsilon}{2}, \quad \text{and} \quad \exp(-\mu h m/4) \leq \frac{\epsilon}{2W_0}\,,$$

which yields

$$h < \frac{\epsilon^2}{24\kappa^2 d^2} \quad \text{and} \quad mh \geq 4/\mu \log\left(2W_0/\epsilon\right)\,.$$

For small enough ϵ, this new h restriction is stronger than (18), and this implies $O(d^2/\epsilon^2 \log(W_0^O/\epsilon))$ iterations are needed. Since each iteration requires only one partial derivative computation, the cost is also $\widetilde{\mathcal{O}}(d^2/\epsilon^2)$. Compared with the classical O-LMC presented in Section 3.2, this cost is in fact $1/\epsilon$ higher.

Remark 4 *While the exponential decay term naturally appears due to the convexity, as was done in the classical analysis for O-LMC, the second term of* (19) *is at the order of* $O(h^{1/2}d)$, *instead of* $O(hd)$ *as in O-LMC. This deterioration comes from the extra error term induced by replacing* $\nabla f(x^m)$ *by* F^m. *As a consequence, the core of the proof lies in bounding*

$$E^m = \nabla f(x^m) - F^m\,. \qquad (20)$$

As shown in Lemma 17, the variance of this term is:

$$\mathbb{E}(|E^m|^2) \lesssim O(d^2)\,.$$

When inserted in the updating formula, this contributes a term of order $O(h^2d^2)$ *and it dominates all other error sources* ($O(h^3d^2)$ *as seen in the classical O-LMC). This eventually leads to a worse estimate. See more detailed discussion in Remark 18.*

Remark 5 *It is a natural question to ask if uniform sampling is necessary. Indeed, in the optimization literature, RCD outperforms GD when the objective function f is skewed, and the coordinates are chosen in a non-uniform way to reflect such skewness. For sampling, we can also use a non-uniform sampling strategy as well.*

To this goal, we first denote the probability of choosing i-th direction ϕ_i and

$$\Phi := \{\phi_1, \phi_2, \dots, \phi_d\}\,, \quad \textit{with} \quad \phi_i > 0\ (\forall i)\,, \quad \textit{and} \quad \sum_{i=1}^{d} \phi_i = 1\,.$$

Then, in the m-th iteration of RCD-LMC, we replace (16) *by*

$$F^m = \frac{1}{\phi_r}\left(\nabla f(x^m)\cdot \boldsymbol{e}^r\right)\boldsymbol{e}^r\,, \tag{21}$$

where x^m is the m-th iteration sample. This is a valid calculation because this definition is still consistent with ∇f in the expectation sense:

$$\mathbb{E}_r\left(\frac{1}{\phi_r}\left(\nabla f(x)\cdot \boldsymbol{e}^r\right)\boldsymbol{e}^r\right) = \nabla f(x)\,.$$

However, using a non-uniform sampling strategy will not provide better convergence than the uniform sampling under our setting. This will be discussed in Remark 19.

4.2.2 Convergence for RCD-U-LMC

In the underdamped setting, we have the following theorem:

Theorem 6 *Suppose f satisfies Assumption 3.1. Set $\gamma = 1/L$. If h satisfies*

$$h < \frac{1}{880 d\kappa}\,, \tag{22}$$

then, we have:

$$W_m \le 4\exp\left(-hm/(8\kappa)\right)W_0 + Cdh^{1/2}\,, \tag{23}$$

where $C = 100(\kappa/\mu)^{1/2}$, $W_m = W_2(q^U_m, p_2)$, and $q^U_m(x,v)$ is the probability density function of (x^m, v^m) computed by RCD-U-LMC.

We discuss the proof in Section 8.2. To obtain ϵ accuracy, we simply set both terms $< \epsilon/2$ in (23), and it yields, in addition to (22):

$$h < \frac{\epsilon^2\mu}{10^4\kappa d^2}\,, \quad \text{and} \quad mh \ge 8\kappa\log\left(8W_0/\epsilon\right)\,.$$

This leads to $O\left(d^2/\epsilon^2\log\left(W_0/\epsilon\right)\right)$ number of iterations, and thus the cost is $\widetilde{\mathcal{O}}(d^2/\epsilon^2)$. Compared with the classical U-LMC as discussed in Section 3.2, this cost is $d^{1/2}/\epsilon$ larger.

Similar to the overdamped case, the estimate for RCD-U-LMC is worse than that for U-LMC due to the newly induced error $E^m = \nabla f(x^m) - F^m$. Once again this term $\mathbb{E}|E^m|^2$ provides a larger error source than other terms ($O(h^3 d)$ as seen in the classical U-LMC), and is, upon the application of the iteration formula, represented in the second term in (23). See more details in Remark 25.

4.3 A counter-example

The two theoretical results above suggest that no improvement is seen when RCD is blindly applied to O/U-LMC, if not worse. We cannot argue the sharpness of the two results, but we do provide a counter-example to demonstrate that including RCD to O/U-LMC may not bring computational advantage.

The example is given to show RCD-U-LMC. We assume

$$q_0(x,v) = \frac{1}{(4\pi)^{d/2}}\exp(-|x-\mathbf{u}|^2/2-|v|^2/2)\,,\quad p_2(x,v) = \frac{1}{(2\pi)^{d/2}}\exp(-|x|^2/2-|v|^2/2)\,, \tag{24}$$

where $\mathbf{u}\in\mathbb{R}^d$ satisfies $\mathbf{u}_i = 1/8$ for all i. Denote $\{(x^m, v^m)\}$ the sample computed through Algorithm 2 (underdamped) with stepsize h. Denote q_m the probability density function of (x^m, v^m), then we can show $W_2(q_m, p_2)$ cannot converge too fast.

Theorem 7 *[Theorem 4.1 in Ding and Li (2020)] For the example above, choose $\gamma = 1$ in Algorithm 2, and let d, h satisfy*

$$d > 1872, \quad h < \frac{1}{1440^2 d}\,,$$

then

$$W_m \geq \exp\left(-2mh\right)\frac{\sqrt{d}}{1024} + \frac{d^{3/2}h}{2304}\,, \tag{25}$$

where $W_m = W_2(q^U_m, p_2)$, and $q^U_m(x,v)$ is the probability density function of (x^m, v^m) computed by RCD-U-LMC.

We note the second term in (25) is rather big. The smallness comes from h, the stepsize, and it needs to be small enough to balance out the influence from $d^{3/2}\gg 1$. This puts a strong restriction on h. Indeed, to have ϵ-accuracy, $W(q^U_m, p_2)\leq\epsilon$, we need both terms smaller than ϵ, and this term suggests that $h\leq\frac{2304\epsilon}{d^{3/2}}$ at least. And when combined with restriction from the first term, we arrive at the conclusion that at least $\widetilde{O}(d^{3/2}/\epsilon)$ iterations are needed, and thus $\widetilde{O}(d^{3/2}/\epsilon)$ partial derivative calculations are required. The d dependence is $d^{3/2}$. This is exactly the same as what U-LMC requires, meaning RCD-U-LMC brings no computational advantage over U-LMC.

Remark 8 *All discussions on convergence for U-LMC methods are conducted without assuming the hessian of f is Lipschitz. A natural question is: If f has a higher regularity, is the method converging faster? The example above gives a negative answer. If we apply the classical U-LMC to this example, the number of iterations to achieve ϵ-accuracy in W_2-distance is at least $\widetilde{O}(d^{1/2}/\epsilon)$. This coincides with the results obtained when only ∇f is assumed to be Lipschitz. For this reason, throughout the paper we do not require high regularity of f under the U-LMC framework.*

5. Variance reduction techniques

Blindly applying RCD to LMC does not bring any numerical savings. In the analysis, the biggest contribution to the large error (the remainder term) is induced by the process of randomly selecting directional derivatives. In this section we propose two variance reduction methods inspired by SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014) and apply them to RCD-O/U-LMC (and thus four algorithms in total). We will prove that while the numerical cost per iteration is reduced by d-folds, the number of iterations needed for achieving the preset accuracy is mostly unchanged compared to the classical LMC. This ultimately saves the total cost.

5.1 Algorithms

5.1.1 SVRG based variance reduction method

As presented in Section 2.4, SVRG is an optimization technique introduced to reduce variance in SGD in its random selection process. Only one representative gradient is computed per iteration, unless the iteration number is an integer times τ, when all gradients are evaluated. It has been proved in (Johnson and Zhang, 2013) that the approach enhances the SGD performance.

Inspired by this method, we propose SVRG-O/U-LMC, as summarized in Algorithm 3. In this algorithm, we compute the full gradient ∇f at $k\tau$ time-step for all integers k, and during the epochs, per iteration, we merely update one directional derivative chosen at random.

Algorithm 3 SVRG-O(U)-LMC

```
**Preparation:**
1. Input: $h$ (time stepsize); $\tau$ (epoch length); $\gamma$ (parameter); $M$ (stopping index); $d$ (dimension) and $f(x)$.
2. Initial: *(overdamped)*: $x^0$ i.i.d. sampled from an initial distribution induced by $q_0(x)$.
    *(underdamped)*: $(x^0, v^0)$ i.i.d. sampled from the initial distribution induced by $q_0(x, v)$.
**Run:** **For** $m = 0\,, 1\,, \cdots\,, M$
    **if** $m$ mod $\tau = 0$ **then** update $\widehat{g}$ and compute flux $F^m$:
        $$F^m_i = \widehat{g}_i = \partial_i f(x^0), \qquad 1 \leq i \leq d\,. \tag{26}$$
    **otherwise**
    Draw a random number $r^m$ uniformly from $1, 2, \cdots, d$ and compute
        $$g^m_{r^m} = \partial_{r^m} f(x^m)\,, \quad F^m = \widehat{g} + d\,(g^m_{r^m} - \widehat{g}_{r^m})\,\mathbf{e}^{r^m}\,. \tag{27}$$
    **end if**
    *(overdamped)*: Draw $\xi^m$ from $\mathcal{N}(0, I_d)$:
        $$x^{m+1} = x^m - F^m h + \sqrt{2h}\xi^m\,. \tag{28}$$
*(underdamped)*: Sample $(x^{m+1}, v^{m+1})$ as Gaussian random variables with expectation and covariance defined in (5), replacing $\nabla f(x^m)$ by $F^m$.
**end**
**Output:** $\{x^m\}$.
```

5.1.2 SAGA based variance reduction method

SAGA is also a technique introduced in optimization to reduce the variance of SGD. It starts with a full gradient, and in each step, simply updates the gradient of one randomly selected representative function.

Inspired by the approach, we propose our Algorithm 4, in which we compute the full gradient only in the first round of iteration, and keep updating one randomly selected directional derivative per iteration. We term the method Random Coordinate Averaging Descent-O/U-LMC (RCAD-O/U-LMC).

Algorithm 4 RCAD-O(U)-LMC

```
Preparation:
1. Input: $h$ (time stepsize); $\gamma$ (parameter); $M$ (stopping index); $d$ (dimension) and $f(x)$.
2. Initial: (overdamped): $x^0$ i.i.d. sampled from the initial distribution induced by $q_0(x)$ and calculate $g^0 \in \mathbb{R}^d$:
        $$g^0_i = \partial_i f(x^0), \quad 1 \le i \le d\,. \qquad (29)$$
   (underdamped): $(x^0, v^0)$ i.i.d. sampled from the initial distribution induced by $q_0(x,v)$ and calculate $g^0 \in \mathbb{R}^d$ as (29).
Run: For $m = 0\,,1\,,\cdots\,,M$
    1. Draw a random number $r^m$ uniformly from $1, 2, \cdots, d$.
    2. Calculate $g^{m+1}$ and flux $F^m \in \mathbb{R}^d$ by letting $g^{m+1}_i = g^m_i$ for $i \neq r^m$ and
        $$g^{m+1}_{r^m} = \partial_{r^m} f(x^m)\,, \quad F^m = g^m + d\left(g^{m+1}_{r^m} - g^m_{r^m}\right)\mathbf{e}^{r^m}\,. \qquad (30)$$
    3. (overdamped): Draw $\xi^m$ from $\mathcal{N}(0, I_d)$:
        $$x^{m+1} = x^m - F^m h + \sqrt{2h}\xi^m\,. \qquad (31)$$
       (underdamped): Sample $(x^{m+1}, v^{m+1})$ as a Gaussian random variable with expectation and covariance defined in (5), replacing $\nabla f(x^m)$ by $F^m$.
end
Output: $\{x^m\}$.
```

5.2 Convergence and numerical cost analysis

We now discuss the convergence of SVRG/RCAD-O-LMC and SVRG/RCAD-U-LMC. In the classical papers (Durmus et al., 2019; Dalalyan and Karagulyan, 2019; Cheng et al., 2018; Dalalyan and Riou-Durand, 2020) that discuss the convergence of O/U-LMC, the authors indeed discussed the numerical error in approximating the gradients, but both papers require that the variance of the error is bounded and independent of d. The random coordinate selection process is rather complicated, making their results hard to apply. One related work is (Chatterji et al., 2018), where the authors construct a Lyapunov function to study the convergence of SG-MCMC. Our proof for the convergence of SVRG/RCAD-O-LMC is inspired by its technicalities. In (Cheng et al., 2018; Chatterji et al., 2018), a contraction map is used for U-LMC, but such a map cannot be directly applied in our situation because the variance depends on the entire trajectory of samples. Furthermore, the history of the trajectory is reflected in each iteration, deeming the process to be non-Markovian. We need to re-engineer the iteration formula accordingly for tracing the error propagation.

5.2.1 Convergence for SVRG/RCAD-O-LMC

For SVRG-O-LMC, We have the following theorem.

Theorem 9 *Suppose f satisfies Assumptions 3.1-3.2 and h satisfies*

$$h < \min\left\{\frac{1}{400d\kappa^2\mu}, \frac{1}{10\tau\max\{\mu,1\}}\right\}. \tag{32}$$

Denote $W_m = W_2(q^O_m, p)$, q^O_m the density of the sample x^m computed from Algorithm 3 (overdamped), and p the target density function, then we have

$$W_m \leq \exp\left(-\frac{\mu h m}{32}\right) W_0 + \left[h^{3/2}\tau d C_1 + h\tau^{1/2} d C_2\right], \tag{33}$$

where

$$C_1 = 30\kappa^{3/2}\mu, \quad C_2 = 50\kappa\sqrt{\mu} + 5\sqrt{\kappa^3\mu/d + 2H^2/\mu^2}\,.$$

We leave the proof to Section 7.3 and simply discuss the numerical cost here. Suppose we set $\tau = d$. And we preset the desired accuracy to be ϵ, meaning we wish to obtain $W_2(q^O_m, p) \leq \epsilon$, then we can simply choose to set all terms in (33) less than $\epsilon/3$. The latter two terms give the constraints to h while the first one gives the constraint on m, meaning in addition to the requirement (32), we have:

$$h \lesssim \min\left\{\frac{\epsilon^{2/3}}{d^{4/3}C_1^{2/3}}, \frac{\epsilon}{d^{3/2}C_2}\right\}, \quad \text{and} \quad mh \gtrsim \frac{1}{\mu}\log\left(\frac{W_0}{\epsilon}\right).$$

For small ϵ and large d, $h < \frac{\epsilon}{d^{3/2}C_2}$ is the most restrictive one, and it leads the cost $m > \widetilde{O}(d^{3/2}/\epsilon)$. Since in most iterations (unless $k\tau$) one only calculates one directional derivative, the total computational cost is $\widetilde{O}(d^{3/2}/\epsilon)$.

For RCAD-O-LMC, We have the following theorem.

Theorem 10 *[Theorem 5.1 in Ding and Li (2020)] Suppose f satisfies Assumptions 3.1-3.2 and h satisfies*

$$h < \frac{1}{3(1+9d)\kappa^2\mu}. \tag{34}$$

Denote $W_m = W_2(q^O_m, p)$, q^O_m the density of the sample x^m computed from Algorithm 4 (overdamped), and p the target density function, then we have

$$W_m \leq \sqrt{2}\exp(-\mu h m/4)W_0 + 2h\sqrt{d^3C_1 + d^2C_2}\,. \tag{35}$$

Here $C_1 = 77\kappa^2\mu$, $C_2 = H^2/\mu^2 + \kappa^3\mu/d$.

We omit the detailed proof for Theorem 10. Interested readers are referred to the appendix of (Ding and Li, 2020). The sketch of the proof is presented in Section 7.4. We here discuss the computational cost. Suppose we present the desired accuracy to be ϵ, then we can simply choose h to set both terms in (35) less than $\epsilon/2$, and it leads to, in addition to (34)

$$h \lesssim \frac{\epsilon}{d^{3/2}\sqrt{C_1 + C_2/d}}\,, \quad \text{and} \quad mh \gtrsim \frac{1}{\mu}\log\left(\frac{W_0}{\epsilon}\right)\,.$$

For small ϵ, the new constraint of h dominates, and this leads to the cost $m > \widetilde{O}(d^{3/2}/\epsilon)$.

We emphasize that in Theorem 9 and Theorem 10 we require both Assumptions 3.1 and 3.2. The second assumption is on the continuity of the hessian term. If this is relaxed, the convergence still holds true, with a degraded numerical cost. The cost will be $\widetilde{O}(\max\{d^{3/2}/\epsilon, d/\epsilon^2\})$. But compared to $\widetilde{O}(d^2/\epsilon^2)$ required by the classical O-LMC, the algorithms with reduced variance still outperform. The proof is the same, and we omit it from the paper.

Remark 11 *We have two comments:*

- *Compare Theorem 3, 9 and 10, the three theorems for overdamped LMC methods, it is rather clear that the dependence on h and m in the exponential is the same. Since m and h have the same order, then for ϵ accuracy, $m \sim \frac{1}{h}$. The h requirement, however, mainly comes from the form of the second term. It is the quality of this term that determines the final numerical cost.*

  In particular, in Theorem 3, h's dependence on d is at the power of -2, but in Theorem 9 and 10, such dependence reduces to $-3/2$. This explains the improvement from RCD to SVRG and RCAD in the overdamped setting, as presented in Table 1.

- *As presented in Remark 4, the major error term that needs to be improved is the second term in (19), which is mainly induced by the variance term $\mathbb{E}|E^m|^2$ that measures the distance between $\nabla f(x^m)$ and F^m. When a variance reduction technique is integrated, we can find a better estimate for $\nabla f(x^m)$, reducing the error $\mathbb{E}|E^m|^2$, and improving the final convergence bound. In particular, as will be shown in Lemma 20 and Lemma 21, we roughly obtain, for SVRG-O-LMC and RCAD-O-LMC respectively,*

  $$\mathbb{E}|E^m|^2 \leq O(h^2d^2\tau^2 + hd^2\tau) \quad \textit{and} \quad \mathbb{E}\,|E^m|^2 \leq O(h^2d^4 + hd^3)\,.$$

  Under the condition $\tau = d$ and $h \lesssim \frac{1}{d}$, both terms, when inserted in the iteration, contribute an error term of order $O(h^3d^3)$. This is an improvement over RCD-O-LMC ($O(h^2d^2)$). More details can be found at the end of Section 7.3 and 7.4.

5.2.2 Convergence for SVRG/RCAD-U-LMC

For SVRG-U-LMC, We have the following theorem.

Theorem 12 *Suppose f satisfies Assumptions 3.1, $\gamma = 1/L$, and h satisfies*

$$h \leq \min\left\{\frac{1}{1648\kappa d}, \frac{1}{40\tau}\right\}\,, \tag{36}$$

then, with $C_1 = 200\sqrt{\frac{\kappa}{\mu}}$ and $C_2 = 240/\sqrt{\mu}$:

$$W_m < 4\exp\left(-\frac{hm}{32\kappa}\right)W_0 + hd^{1/2}C_1 + h^{3/2}\tau dC_2\,. \tag{37}$$

Here $W_m = W_2(q^U_m, p_2)$, and q^U_m denotes the density of the sample (x^m, v^m) derived from Algorithm 3 (underdamped), and p_2 is the target density function.

We leave the proof to Section 8.3. The theorem gives us the strategy of choosing parameters: to achieve ϵ-accuracy, meaning to have $W_m \leq \epsilon$, we can set all terms in (37) less than $\epsilon/3$, and it leads to, when $\tau = d$, in addition to (36)

$$h \lesssim \min\left\{\frac{\epsilon}{d^{1/2}C_1}, \frac{\epsilon^{2/3}}{d^{4/3}C_2^{2/3}}\right\}, \quad \text{and} \quad mh \gtrsim \kappa\log\left(\frac{4W_0}{\epsilon}\right).$$

It is hard to balance the smallness of ϵ and the largeness of d, and we keep both restrictions, this means $m > \widetilde{O}\left(\max\left\{\frac{d^{4/3}}{\epsilon^{2/3}}, \frac{d^{1/2}}{\epsilon}\right\}\right)$.

For RCAD-U-LMC, we have the following theorem:

Theorem 13 *[Theorem 5.2 in Ding and Li (2020)] Assume $f(x)$ satisfies Assumption 3.1, $\gamma = 1/L$, and h satisfies*

$$h \leq \frac{1}{1648\kappa d}\,, \tag{38}$$

then, with $C_1 = 200\sqrt{\frac{\kappa}{\mu}}$ and $C_2 = 200/\sqrt{\mu}$:

$$W_m \leq 4\sqrt{2}\exp\left(-\frac{hm}{8\kappa}\right)W_0 + hd^{1/2}C_1 + h^{3/2}d^2C_2\,, \tag{39}$$

where $W_m = W_2(q^U_m, p_2)$ and q^U_m is the density of the sample (x^m, v^m) derived from Algorithm 4 (underdamped).

The proof of this theorem is referred to (Ding and Li, 2020), and we sketch the main strategy in Section 8.4. Here we only discuss computational cost. To achieve ϵ-accuracy, meaning to have $W_m \leq \epsilon$, we can choose all terms in (39) less than $\epsilon/3$. This gives, in addition to (38).

$$h \lesssim \min\left\{\frac{\epsilon}{d^{1/2}C_1}, \frac{\epsilon^{2/3}}{d^{4/3}C_2^{2/3}}\right\}, \quad \text{and} \quad mh \gtrsim \kappa\log\left(\frac{4\sqrt{2}W_0}{\epsilon}\right).$$

For small ϵ and large d, the second term in the h bound is smaller than the third term, and this means $m > \widetilde{O}\left(\max\left\{\frac{d^{4/3}}{\epsilon^{2/3}}, \frac{d^{1/2}}{\epsilon}\right\}\right)$.

Remark 14 *The improvement observed in the underdamped case is an analogy to that seen for the overdamped case. By integrating variance reduction techniques, we have a better estimate of $\nabla f(x^m)$, and this reduces the cost of $\mathbb{E}|E^m|^2$. The control of this term is presented in Lemma 26 and Lemma 27 respectively. More detailed discussion can be found at the end of Section 8.3 and 8.4.*

6. Numerical result

In this section, we test the efficiency of variance reduction enhanced RCD methods using the following three examples. We should mention that it is numerically challenging to compute the Wasserstein distance on a high dimensional space, so we present the convergence of error only in the weak sense, by testing it on a test function.

- Example 1: In this example, our target distribution is $\mathcal{N}(0, I_d)$ with $d = 1000$, meaning

$$p(x) \propto \exp\left(-\sum_{i=1}^{d} \frac{|x_i|^2}{2}\right).$$

The initial distributions, for the overdamped and underdamped situations respectively, are $\mathcal{N}(\mathbf{0.5}, I_d)$ and $\mathcal{N}(\mathbf{0.5}, I_{2d})$. We run RCD-O/U-LMC, RCAD-O/U-LMC and SVRG-O/U-LMC with $N = 5 \times 10^5$ particles with different stepsizes h, and test expectation estimation error by calculating

$$\text{Error} = \left|\frac{1}{N}\sum_{i=1}^{N} \phi(x^i) - \mathbb{E}_p(\phi)\right|, \tag{40}$$

where ϕ is the test function. In Figure 1, we show the error with $\phi(x) = |x_1|^2$ using different stepsizes. In both underdamped and overdamped cases, the improvement of convergence rate is immediate when variance reduction techniques are added.

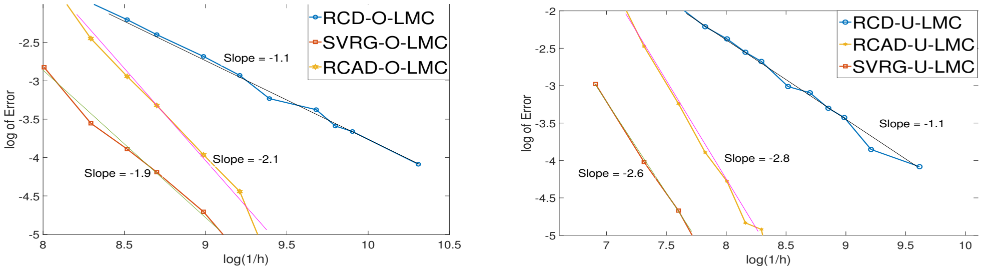

Figure 1: Decay of Error of LMC in overdamped (left) and underdamped (right) settings.

- Example 2: In this example, our target distribution is the summation of two Gaussians whose density is

$$p(x) \propto \exp\left(-\sum_{i=1}^{d} \frac{|x_i - 2|^2}{2}\right) + \exp\left(-\sum_{i=1}^{d} \frac{|x_i + 2|^2}{2}\right).$$

This target distribution has two peaks and does not satisfy the assumptions in this paper. However, the experimental results still suggest the outperformance when variance reduction is imposed. We choose $d = 1000$ and $N = 10^6$ particles. The initial distributions, for the overdamped and underdamped situations respectively, are $\mathcal{N}(\mathbf{0}, I_d)$ and $\mathcal{N}(\mathbf{0}, I_{2d})$. We run RCD-O/U-LMC, RCAD-O/U-LMC, SVRG-O/U-LMC with

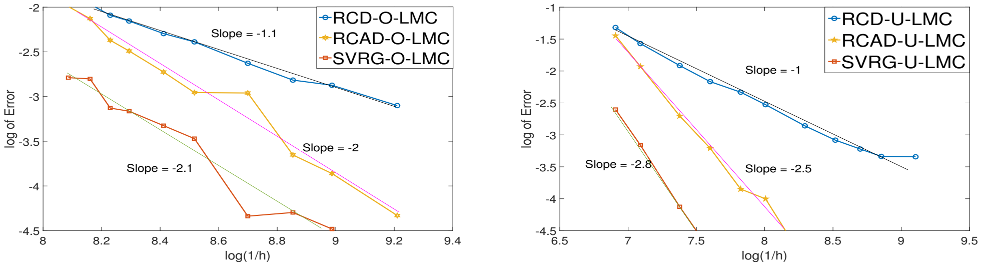

Figure 2: Decay of Error of LMC in overdamped (left) and underdamped (right) settings.

different h, and calculate error using (40) with $\phi(x) = |x_1|^2$. In both underdamped and overdamped settings, including variance reduction techniques provide better accuracy, as shown in Figure 2.

- Example 3 (Generalized linear regression) : In this example, we consider the following linear regression problem:

  $$b = x \cdot a + \eta\,, \tag{41}$$

  where $a, x \in \mathbb{R}^d$ and η, the noise is a real-valued random variable satisfying $p_{\mathrm{nos}}(\eta)$. Suppose $p_{\mathrm{nos}}(\eta)$ is known, and one is given a set of data $\{(a_i, b_i)\}_{i=1}^I$ to infer x. According to the Bayes' rule, given the prior distribution of x, termed p_{prior}, the posterior distribution, also our target distribution is:

  $$p(x) = p_{\mathrm{pos}}(x) \propto p_{\mathrm{prior}}(x)\Pi_{i=1}^I p_{\mathrm{nos}}(b_i - x \cdot a_i)\,. \tag{42}$$

  Now we look for samples sampled according to p_{pos}. In the test, we choose $d = 100$, $x = \mathbf{1}$ and generate $I = 100$ data pairs. We further set the prior to be $\mathcal{N}(\mathbf{0}, I_d)$, and we evaluate the algorithms with two different choices of p_{nos}. In the first choice, this noise distribution is $\mathcal{N}(0,1)$, and in the second choice, we set:

  $$p_{\mathrm{nos}}(\eta) \propto \exp\left(-\frac{|\eta|^2 + \cos(\eta)}{2}\right)\,. \tag{43}$$

  All six algorithms, RCD-O/U-LMC, RCAD-O/U-LMC, SVRG-O/U-LMC are run with $N = 10^6$. The error is calculated using (40) with $\phi(x) = |\mathsf{x}|^2$, where $\mathsf{x} = (x_1, x_2, \ldots, x_{10})$ is the first ten components of x.

  In Figure 3 and 4, η satisfies standard Gaussian and (43) respectively. In both cases, the variance reduction methods converge faster.

In all the numerical examples above, we observe, in the O-LMC framework, the classical O-LMC gives the error saturating at $\sim h$ while the variance reduced algorithms roughly gives saturation error $\sim h^2$. In the U-LMC framework, the dependence of such error on h increases from 1 to ~ 2.5 when variance reduction techniques are incorporated. We should stress that we numerically observe better convergence rates than those stated in the theorems. This is potentially due to the fact that we are evaluating the error in the weak sense by testing the samples on a test function. This is a weaker criterion than the Wasserstein distance discussed in the theorems.

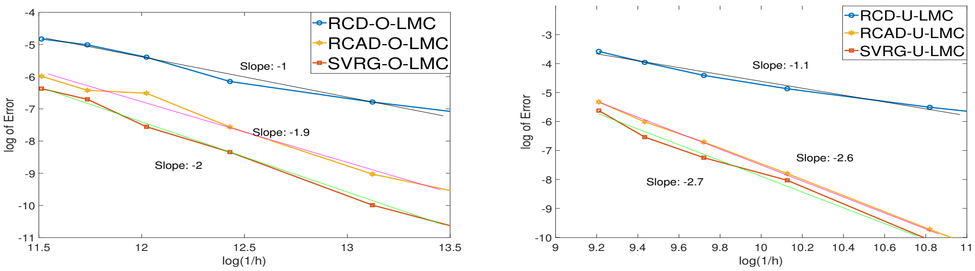

Figure 3: Decay of Error of LMC in overdamped (left) and underdamped (right) settings.

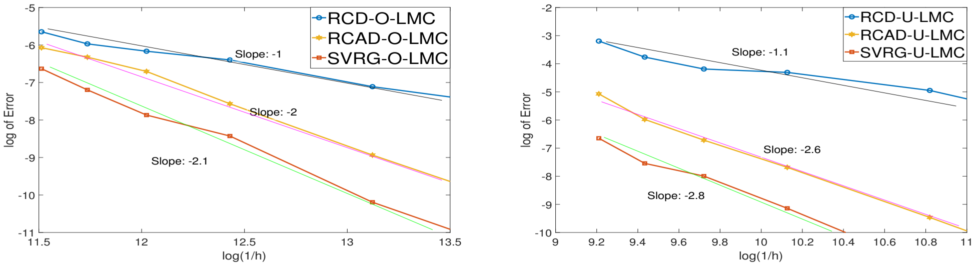

Figure 4: Decay of Error of LMC in overdamped (left) and underdamped (right) settings.

7. Proof of convergence results for O-LMC

We analyze the three O-LMC methods (RCD-O-LMC, SVRG-O-LMC, and RCAD-O-LMC) together in this section.

Before diving into details, we quickly summarize the proving strategy. Recall that the target distribution p is merely the equilibrium of the SDE (1). This means, if a particle prepared at the initial stage is drawn from p, then following the dynamics of SDE (1), the distribution of this particle will continue to be p. In the analysis below, we call the trajectory of this particle y_t, and the sequence generated by this particle evaluated at discrete time y^m. Essentially we evaluate how quickly x^m converges to y^m as m increases. In particular, we call $\Delta^m = x^m - y^m$ and will derive an iteration formula that shows the convergence of Δ^m.

In evaluating Δ^m, two kinds of error get involved:

1. discretization error in h: this amounts to controlling the discretization error of the SDE (1). To handle this part of the error we employ the estimates in (Dalalyan and Karagulyan, 2019);

2. random coordinate selection process error: this is to measure, at each iteration, how big can $\nabla f(x^m) - F^m$ be. According to the way F^m is defined, it is straightforward to show that the expectation of this error is always 0, but the variance $\mathbb{E}|\nabla f(x^m) - F^m|^2$ can be big. The analysis for the three different methods under O-LMC framework mostly concentrates on giving a bound to this term.

In a way, see details in (54), we can derive the iteration formula:

$$\begin{aligned}\Delta^{m+1} =& y^{m+1} - x^{m+1} = \Delta^m + (y^{m+1} - y^m) - (x^{m+1} - x^m)\\ =& \Delta^m - h\left(\nabla f(y^m) - \nabla f(x^m)\right) - \int_{mh}^{(m+1)h} \left(\nabla f(y_s) - \nabla f(y^m)\right) \mathrm{d}s\\ & - h(\nabla f(x^m) - F^m)\,.\end{aligned}$$

The second term on the right-hand side, by using the Lipschitz continuity, will provide $-Lh\Delta^m$, and it produces desirable properties when combined with the first Δ^m. The third term encodes the discretization error in h and was shown to be small in (Dalalyan and Karagulyan, 2019). The last term is the error that comes from the random coordinate selection process, and different methods will give different control.

It turns out that all three O-LMC methods share the same iterative formula for Δ^m. We will derive this formula in Theorem 15. The formula, however, requires the boundedness of a variance term $\mathbb{E}|\nabla f - F|^2$, and all three methods will provide different bounds. In subsection 7.1, we discuss the background and derive the iteration formula. The subsequent three subsections are dedicated to the analysis of the variance term. When combined with Theorem 15, the theorems are immediate.

7.1 Iterative formula for Δ^m

We first rewrite (17), (28), (31) into

$$x^{m+1} = x^m - \nabla f(x^m)h + E^m h + \sqrt{2h}\xi^m\,, \tag{44}$$

where E^m is defined in (20):

$$E^m = \nabla f(x^m) - F^m\,.$$

It can be shown that in all three cases, if r^m is chosen uniformly from $1\,, \cdots, d$:

$$\mathbb{E}_{r^m}(E^m) = \mathbf{0}\,. \tag{45}$$

We simply need to recall the definition of F^m. In RCD, we have:

$$F^m = d\partial_{r^m} f(x^m)\mathbf{e}^{r^m}\,, \tag{46}$$

and for SVRG-O-LMC it becomes:

$$F^m = \widehat{g} + d\left(g^m_{r^m} - \widehat{g}_{r^m}\right)\mathbf{e}^{r^m}\,, \quad \text{with} \quad g^m_{r^m} = \partial_{r^m} f(x^m),\ \widehat{g} = \nabla f\left(x^{\lfloor \frac{m}{\tau} \rfloor}\right)\,, \tag{47}$$

and for RCAD-O-LMC, it is:

$$F^m = g^m + d\left(g^{m+1}_{r^m} - g^m_{r^m}\right)\mathbf{e}^{r^m}\,, \quad \text{with} \quad g^{m+1}_{r^m} = \partial_{r^m} f(x^m)\,. \tag{48}$$

Taking the expectation with respect to r^m, (45) is straightforward.

We then define the dynamics of y_t to be:

$$y_t = y^0 - \int_0^t \nabla f(y_s)\,\mathrm{d}s + \sqrt{2}\int_0^t \mathrm{d}B_s\,, \tag{49}$$

then, calling $y^m = y_{hm}$, and letting $B_{h(m+1)} - B_{hm} = \sqrt{h}\xi^m$, we have:

$$y^{m+1} = y^m - \int_{mh}^{(m+1)h} \nabla f(y_s)\,\mathrm{d}s + \sqrt{2h}\xi^m\,. \tag{50}$$

Assume y_0 is a random vector drawn from the target distribution induced by p such that $W_2^2(q_0^O, p) = \mathbb{E}|x^0 - y^0|^2$, we have y_t is drawn from the distribution induced by p for all t.

Denote

$$\Delta^m = y^m - x^m\,, \tag{51}$$

then

$$W_2^2(q_m^O, p) \leq \mathbb{E}|\Delta^m|^2 = \mathbb{E}|x^m - y^m|^2\,,$$

where $\mathbb{E}$ takes all randomness into account. We now essentially need to give a bound to Δ^m by comparing (44) and (50).

In Theorem 15 we derive the iteration formula for updating Δ^m.

Theorem 15 *Suppose f satisfies Assumptions 3.1-3.2, if $\{x^m\}$ is defined in (44), $\{y^m\}$ is defined in (50) and $\{\Delta^m\}$ comes from (51), then for any $a > 0$ and $m \geq 0$, we have*

$$\mathbb{E}|\Delta^{m+1}|^2 \leq (1+a)A\mathbb{E}|\Delta^m|^2 + 3(1+a)h^2\mathbb{E}|E^m|^2 + (1+a)h^3B + \left(1+\frac{1}{a}\right)h^4C\,, \tag{52}$$

where

$$A = 1 - 2\mu h + 3L^2h^2\,,\quad B = 2L^2d\,,\quad C = \frac{1}{2}\left(H^2d^2 + L^3d\right)\,. \tag{53}$$

The only formula this lemma requires is (44). There is no assumption added on E^m besides $\mathbb{E}_{r^m}E^m = 0$, and thus this result is applicable to all three algorithms. However, as discussed before, the variance of E^m of these three methods are significantly different. The following three subsections are then dedicated to analyzing this variance term in each different method. These, together with Theorem 15, finally conclude the three theorems for overdamped LMC.

Remark 16 *We carefully observe the relations of the four terms in (52).*

- *The coefficient in front of $\mathbb{E}|\Delta^m|^2$ is $(1+a)A$ where $A \sim 1-h$. Suppose a is chosen to be small enough to make $(1+a)A < 1$ strictly, then we have the hope of having exponential decay. This means a should be of the order of $O(h)$.*
- *Then we need to handle the remaining three terms. The last two terms come from translating the SDE (1) to O-LMC. This discretization error cannot be removed, and they are of $O(h^3d^2)$ (since $1/a \sim 1/h$ according to the previous comment). In the ideal case, we would like to have $\mathbb{E}|E^m|^2$ (the variance) term to be negligible. If so, upon the iteration formula, one h drops out, and with taking the square root, the final version of the formula should be in line with $W_m \leq \exp\left(-hm\right)W_0 + dh$. This would lead to the estimate of $m = O(\frac{d}{\epsilon})$ for an ϵ accuracy. In the overdamped setting, this would be the best possible sampling scheme one could hope for when we assume the variance term does not contribute much error.*

Proof We first divide Δ^{m+1} into several parts:

$$\begin{aligned}\Delta^{m+1} =&\Delta^m + (y^{m+1} - y^m) - (x^{m+1} - x^m)\\
=&\Delta^m + \left(-\int_{mh}^{(m+1)h} \nabla f(y_s)\,\mathrm{d}s + \sqrt{2h}\xi_m\right) - \left(-\int_{mh}^{(m+1)h} F^m\,\mathrm{d}s + \sqrt{2h}\xi_m\right)\\
=&\Delta^m - \left(\int_{mh}^{(m+1)h} \left(\nabla f(y_s) - F^m\right)\,\mathrm{d}s\right)\\
=&\Delta^m - \left(\int_{mh}^{(m+1)h} \left(\nabla f(y_s) - \nabla f(y^m) + \nabla f(y^m) - \nabla f(x^m) + \nabla f(x^m) - F^m\right)\,\mathrm{d}s\right)\\
=&\Delta^m - h\left(\nabla f(y^m) - \nabla f(x^m)\right) - \int_{mh}^{(m+1)h} \left(\nabla f(y_s) - \nabla f(y^m)\right)\,\mathrm{d}s\\
&- h(\nabla f(x^m) - F^m)\,.\end{aligned} \tag{54}$$

Note that here $\nabla f(y^m) - \nabla f(x^m)$ can be bounded by Δ^m using the Lipschitz condition, the term $\int \left(\nabla f(y_s) - \nabla f(y^m)\right)\,\mathrm{d}s$ reflects the numerical error accumulated in one-step of SDE computation. $\nabla f(x^m) - F^m$ is the E^m term.

Denote

$$\begin{aligned}U^m &= \nabla f(y^m) - \nabla f(x^m)\,,\\
V^m &= \int_{mh}^{(m+1)h} \left(\nabla f(y_s) - \nabla f(y^m) - \sqrt{2}\int_{mh}^{s} \nabla^2 f(y_r)\,\mathrm{d}B_r\right)\,\mathrm{d}s\,,\\
\Phi^m &= \frac{\sqrt{2}}{h}\int_{mh}^{(m+1)h}\int_{mh}^{s} \nabla^2 f(y_r)\,\mathrm{d}B_r\,\mathrm{d}s\,,\end{aligned}$$

we now have

$$\begin{aligned}\Delta^{m+1} =&\Delta^m - hU^m - (V^m + h\Phi^m) - hE^m\\
=&\Delta^m - V^m - h(U^m + \Phi^m + E^m)\,.\end{aligned} \tag{55}$$

Upon getting equation (55) it is time to analyze each term and hopefully derive an induction inequality that states $\mathbb{E}|\Delta^{m+1}|^2 \approx c\mathbb{E}|\Delta^m|^2 + d$ with $c < 1$ and d being of high orders of h, the parameters we can tune. According to Lemma 6 of (Dalalyan and Karagulyan, 2019), we first have

$$\mathbb{E}|V^m|^2 \leq \frac{h^4}{2}\left(H^2d^2 + L^3d\right)\,,\quad \mathbb{E}|\Phi^m|^2 \leq \frac{2L^2hd}{3}\,, \tag{56}$$

and thus we have:

$$\begin{aligned}&\mathbb{E}|U^m + \Phi^m + E^m|^2\\
=&3\mathbb{E}|U^m|^2 + 3\mathbb{E}|\Phi^m|^2 + 3\mathbb{E}|E^m|^2\,,\\
\leq&3L^2\mathbb{E}|\Delta^m|^2 + 2L^2hd + 3\mathbb{E}|E^m|^2\end{aligned} \tag{57}$$

where we used the Lipschitz continuity of f for controlling U^m, and (56) for Φ^m.

We then handle the cross terms. For example, due to the independence, (45), and the convexity, we have:

$$\mathbb{E}\left\langle\Delta^m, \Phi^m\right\rangle = 0\,,\quad \mathbb{E}\left\langle\Delta^m, E^m\right\rangle = 0\,,\quad \left\langle\Delta^m, U^m\right\rangle \geq \mu|\Delta^m|^2\,, \tag{58}$$

this means the cross term between first and the third term in the last line (55) leads to $-2Mh\mathbb{E}|\Delta^m|^2$. The cross term produced by the first and the second term, however, can be hard to control, mostly because $\mathbb{E}\left<\Delta^m, V^m\right>$ is unknown. We now employ Young's inequality, meaning, for any $a>0$:

$$\mathbb{E}|\Delta^{m+1}|^2 \leq (1+a)\mathbb{E}|\Delta^{m+1}+V^m|^2 + \left(1+\frac{1}{a}\right)\mathbb{E}|V^m|^2\,. \tag{59}$$

While the second term is already bounded in (56), the first term of (59), according to (54) becomes:

$$\begin{aligned}\mathbb{E}|\Delta^{m+1}+V^m|^2 =&\mathbb{E}|\Delta^m - h(U^m+\Phi^m+E^m)|^2\\ =&\mathbb{E}|\Delta^m|^2 - 2h\mathbb{E}\left<\Delta^m, U^m+\Phi^m+E^m\right>\\ &+h^2\mathbb{E}|U^m+\Phi^m+E^m|^2\\ \leq&(1-2\mu h)\mathbb{E}|\Delta^m|^2 + h^2\mathbb{E}|U^m+\Phi^m+E^m|^2\end{aligned}\,, \tag{60}$$

where we used (58). Plug (57) into (60), using the definition of the coefficients A, B in (53):

$$\mathbb{E}|\Delta^m - h(U^m+\Phi^m+E^m)|^2 \leq A\mathbb{E}|\Delta^m|^2 + 3h^2\mathbb{E}|E^m|^2 + h^3B\,. \tag{61}$$

Finally, we plug (56) and (61) into (59) and use the definition of the coefficients C in (53). Then we conclude (52). ■

7.2 Proof of Theorem 3

As discussed before, the iteration formula for Δ^m in RCD-O-LMC satisfies the formula in Theorem 15, and thus the key is to give a bound to $\mathbb{E}|E^m|^2$ for RCD-O-LMC. For that, we use the following lemma.

Lemma 17 *Suppose f satisfies Assumption 3.1,3.2, if $\{x^m\}$ is defined in (17) by RCD-O-LMC, $\{y^m\}$ is defined in (50) and $\{\Delta^m\}$ comes from (51), then for all $m \geq 0$:*

$$\mathbb{E}(|E^m|^2) < 2dL^2\mathbb{E}|\Delta^m|^2 + 2d^2L\,. \tag{62}$$

The proof is shown in Appendix A. Now we are ready to prove Theorem 3.

Proof [Proof of Theorem 3] First, we prove (19). Plug (62) into (52), we have

$$\mathbb{E}|\Delta^{m+1}|^2 \leq (1+a)A'\mathbb{E}|\Delta^m|^2 + (1+a)(6Ld^2h^2+h^3B) + \left(1+\frac{1}{a}\right)h^4C\,,$$

where B and C are kept untouched as in (53) and

$$A' = 1-2\mu h + 3L^2h^2 + 6L^2dh^2\,.$$

Noting $A' < 1-\mu h$ since

$$h < \frac{1}{3\kappa^2\mu + 6\kappa^2\mu d} = \frac{\mu}{3L^2+6L^2d}\,.$$

To ensure the decay of $\mathbb{E}|\Delta^m|^2$, we need to choose a such that $(1+a)A' = 1 - O(h)$. Set $a = \frac{\mu h/2}{1-\mu h}$ so that

$$(1+a)(1-\mu h) = 1 - \frac{\mu h}{2}\,, \quad \text{and} \quad 1 + 1/a \leq 2/\mu h\,,$$

then we have

$$\mathbb{E}|\Delta^{m+1}|^2 \leq \left(1 - \frac{\mu h}{2}\right)\mathbb{E}|\Delta^m|^2 + 2h^2\left(6Ld^2 + hB + hC/\mu\right)\,, \tag{63}$$

where we use $1 + a \leq 2$. Using (63) iteratively, we finally have

$$\mathbb{E}|\Delta^m|^2 \leq \left(1 - \frac{\mu h}{2}\right)^m \mathbb{E}|\Delta^0|^2 + 32h\kappa d^2 \leq \exp\left(-\frac{\mu h m}{2}\right)\mathbb{E}|\Delta^0|^2 + 32h\kappa d^2\,, \tag{64}$$

where we use

$$hB/\mu \leq \kappa d^2, \quad hC/\mu^2 \leq \kappa d^2$$

by $h < \min\left\{\frac{d}{2\kappa\mu}, \frac{2}{H^2/(\kappa\mu^2)+\kappa^2\mu/d}\right\}$. Since

$$(W_m^O)^2 \leq \mathbb{E}|\Delta^m|^2, \quad \mathbb{E}|\Delta^0|^2 = (W_0^O)^2\,,$$

we have (19). ■

Remark 18 *We comment that in the proof, the discretization error terms B and C contribute $O(h^3d^2)$ but the newly induced $\mathbb{E}|E|^2$ term from (62) contributes $O(h^2d^2)$, which eventually becomes the dominating term in (19) aside from the exponential decaying term. This is the term that suggests that the number of required iterations has to be at the order of d^2.*

Remark 19 *As mentioned in Remark 5, using non-uniform sampling to select coordinates may not enhance the computation, when f has moderate conditioning. We provide the intuition here.*

When f is not skewed, $\kappa = L/\mu \sim 1$. Without loss of generality, we assume f attains its minimal at origin, then:

$$\mu|x|^2 \leq |f(x)| \leq L|x|^2 \Rightarrow |f(x)| \sim |x|^2\,, \tag{65}$$

and for every $1 \leq i \leq d$

$$\mu|x_i|^2 \leq |\partial_i f(x)|^2 \leq L|x_i|^2 \Rightarrow |\partial_i f(x)|^2 \sim |x_i|^2\,.$$

When m is large, the distribution of x^m is close to the target distribution. For simplicity, we further assume $x^m \sim p$. The using (21), we calculate the variance of E^m that is defined in (20):

$$\mathbb{E}_r\left(\left|\frac{1}{\phi_r}\left(\nabla f(x^m)\cdot \boldsymbol{e}^r\right)\boldsymbol{e}^r - \nabla f(x^m)\right|^2\right) = \sum_{i=1}^d \left(\frac{1}{\phi_i} - 1\right)\mathbb{E}|\partial_i f(x^m)|^2 \sim \sum_{i=1}^d \left(\frac{1}{\phi_i} - 1\right)\mathbb{E}_p(|x_i^m|^2)\,.$$

According to (65), $\mathbb{E}_p(|x_i|^2) \sim 1$, *and thus the derivation above becomes:*

$$\mathbb{E}\left(\left|\frac{1}{\phi_r}\left(\nabla f(x^m)\cdot \boldsymbol{e}^r\right)\boldsymbol{e}^r - \nabla f(x^m)\right|^2\right) \sim \sum_{i=1}^{d}\left(\frac{1}{\phi_i}-1\right). \tag{66}$$

Noting $\sum_{i=1}^d \phi_i = 1$, *the right-hand side of* (66) *achieves the smallest value when* $\phi_i = 1/d$. *This recovers the uniform RCD-LMC.*

This discussion suggests that the non-uniform selection of coordinates does not bring computational efficiency to RCD-LMC when f has moderate conditioning. We note that the same scenario is observed in optimization (Nesterov, 2012) where the outperformance of RCD over GD crucially depends on the skewness of f. When f is skewed, then the Lipschitz constants in different directions are drastically different. We investigate the conditioning's effect on how coordinates are chosen in (Ding et al., 2021b,a), where we show improvement can be observed if stiffer directions are chosen more frequently.

7.3 Proof of Theorem 9

The strategy for showing SVRG-O-LMC is the same. We first give a bound to the variance term, and then combine the estimate with Theorem 15 for proving the theorem.

Lemma 20 *Suppose f satisfies Assumptions 3.1-3.2, if $\{x^m\}$ is defined in* (28) *by SVRG-O-LMC, $\{y^m\}$ is defined in* (50) *and $\{\Delta^m\}$ comes from* (51), *then for all $m \geq 0$, define*

$$s = \left\lfloor \frac{m}{\tau} \right\rfloor,$$

then we have

$$\mathbb{E}\,|E^m|^2 \leq 3dL^2\left[\mathbb{E}\,|\Delta^m|^2 + \mathbb{E}\,|\Delta^{s\tau}|^2 + 2h^2Ld\tau^2 + 4hd\tau\right]. \tag{67}$$

We put the proof in Appendix B. Now we are ready to prove Theorem 9
Proof [Proof of Theorem 9] Recall

$$s = \left\lfloor \frac{m}{\tau} \right\rfloor.$$

Use (52), similar to (63), set $a = \frac{\mu h/2}{1-\mu h}$, use $h < \frac{1}{48d\kappa^2\mu} < \frac{\mu}{3L^2}$, we have

$$1 + a = \frac{1-\mu h/2}{1-\mu h} < 2, \quad 1 + 1/a = 2/(\mu h),$$

then

$$\begin{aligned}\mathbb{E}|\Delta^{m+1}|^2 &\leq \left(1-\frac{\mu h}{2}\right)\mathbb{E}|\Delta^m|^2 + 6h^2\mathbb{E}|E^m|^2 + 2h^3B + 2h^3C/\mu \\ &\leq \left(1-\frac{\mu h}{2}\right)\mathbb{E}|\Delta^m|^2 + 6h^2\mathbb{E}|E^m|^2 + h^3D,\end{aligned} \tag{68}$$

where we set $D = 2B + 2C/\mu$.

Plug (67) into (68),

$$\begin{aligned}\mathbb{E}|\Delta^{m+1}|^2 \le \left(1-\frac{\mu h}{4}\right)\mathbb{E}|\Delta^{m}|^2 + 18h^2dL^2\mathbb{E}\left|\Delta^{s\tau}\right|^2 \\ + 36h^4\tau^2d^2L^3 + 72\tau d^2h^3L^2 + h^3D\,,\end{aligned} \tag{69}$$

where we use $h < \frac{1}{72d\kappa^2\mu} = \frac{\mu}{72dL^2}$.

Use (69) iteratively from $s\tau$ to m, we have

$$\begin{aligned}\mathbb{E}|\Delta^{m}|^2 \le \left(1-\frac{\mu h}{4}\right)^{m-s\tau}\mathbb{E}|\Delta^{s\tau}|^2 + 18h^2(m-s\tau)dL^2\mathbb{E}\left|\Delta^{s\tau}\right|^2 \\ + 36h^4\tau^3d^2L^3 + 72h^3\tau^2d^2L^2 + h^3\tau D\,.\end{aligned} \tag{70}$$

Since $\mu h\tau < 10$, we have $(1-\frac{\mu h}{4})^\tau < 1-\frac{\mu h\tau}{8}$, and with the other $h^2\tau dL^2$ term, this is still controlled by $1-\frac{\mu h\tau}{16}$. Therefore, choose $m=(s+1)\tau$ and use $h<\frac{\mu}{400dL^2}$, we have

$$\begin{aligned}\mathbb{E}|\Delta^{(s+1)\tau}|^2 &\le \left(1-\frac{\mu h\tau}{16}\right)\mathbb{E}|\Delta^{s\tau}|^2 + 36h^4\tau^3d^2L^3 + 72h^3\tau^2d^2L^2 + h^3\tau D \\ &< \left(1-\frac{\mu h\tau}{16}\right)^{s+1}\mathbb{E}|\Delta^{0}|^2 + 576h^3\tau^2d^2L^3/\mu + 1152h^2\tau d^2L^3/\mu + 16h^2D/\mu\,.\end{aligned} \tag{71}$$

Plug (71) back into (70), use $18h^2\tau dL^2<1$, we have

$$\begin{aligned}\mathbb{E}|\Delta^{m}|^2 < \left(\left(1-\frac{\mu h}{4}\right)^{m-s\tau} + 18h^2\tau dL^2\right)\left(1-\frac{\mu h\tau}{16}\right)^{s}\mathbb{E}|\Delta^{0}|^2 \\ + 650h^3\tau^2d^2L^3/\mu + 1250h^2\tau d^2L^2/\mu + 17h^2D/\mu\,.\end{aligned}$$

Use $h\tau<\frac{1}{10}$ and $18h^2(m-s\tau)dL^2 < \frac{\mu h(m-s\tau)}{16}$, we have

$$\begin{aligned}\left(1-\frac{\mu h}{4}\right)^{m-s\tau} + 18h^2(m-s\tau)dL^2 \le 1-\frac{\mu h(m-s\tau)}{8} + 18h^2(m-s\tau)dL^2 \le 1-\frac{\mu h(m-s\tau)}{16} \\ \le \left(1-\frac{\mu h}{16}\right)^{m-s\tau}\,,\end{aligned}$$

which implies

$$\mathbb{E}|\Delta^{m}|^2 < \exp\left(-\frac{\mu hm}{16}\right)\mathbb{E}|\Delta^{0}|^2 + 650h^3\tau^2d^2L^3/\mu + 1250h^2\tau d^2L^2/\mu + 17h^2D/\mu\,. \tag{72}$$

Taking square root on both sides and use

$$D = 2B + 2C/\mu \le d^2(4\kappa^2\mu^2/d + \kappa^3\mu^2/d + 2H^2/\mu)\,.$$

Since $\tau \ge 1, d \ge 1$, combine last two terms in (72), we have (33). ■

Similar to the comment at the end of Section 7.2, we here briefly review the d and h dependence. As is the case in Section 7.2, the B term and the C term (glued together to be called the D term here) contribute h^3. The error induced by $\mathbb{E}|E|^2$ is bounded by (67), when entering (70) is also at the order of h^3. These two terms combined explain the last term in (33). However we should note that τ is typically set at the order of d, so the d dependence from the newly introduced error in (71) is in fact d^3 as a comparison of d^2 in D. This d^3 term determines the $m \sim d^{3/2}$ dependence in the end.

7.4 Sketch proof of Theorem 10

The same strategy as was used in the previous two subsections is used here to give an estimate to RCAD-O-LMC. We first bound the variance term $\mathbb{E}|E^m|^2$.

We first define a few notations. Define β^m using (48) with the same r^m but replacing x^m with y^m:

$$\beta^0 = \nabla f(y^0) \tag{73}$$

and

$$\beta^{m+1}_{r^m} = \partial_{r^m} f(y^m) \quad \text{and} \quad \beta^{m+1}_i = \beta^m_i \quad \text{if} \quad i \neq r^m\,. \tag{74}$$

Then we have:

Lemma 21 *Suppose f satisfies Assumptions 3.1-3.2, if $\{x^m\}$ is defined in (31) by RCAD-O-LMC, $\{y^m\}$ is defined in (50), and $\{\Delta^m\}$ comes from (51), then for all $m \geq 0$, we have*

$$\mathbb{E}\left|E^m\right|^2 \leq 3dL^2\mathbb{E}|\Delta^m|^2 + 24hL^2d^3\left[hLd+1\right] + 3d\mathbb{E}\left|\beta^m - g^m\right|^2\,. \tag{75}$$

The proof is seen in (Ding and Li, 2020). One big difference between this analysis and the counterparts in the previous two subsections is that the bound of $\mathbb{E}|E^m|^2$ depends on $\nabla f(y^m) - \nabla f(x^m)$, reflected in the $\beta^m - g^m$ term in (75). This term traces the history of the random coordinate selection, and cannot be merely bounded by Δ^m, making the full iteration self-consistent. To overcome the difficulty, we define a Lyapunov function that combines the effect of Δ^m and this particular term $\beta^m - g^m$.

The Lyapunov function is defined as follows:

$$T^m = T^m_1 + c_p T^m_2 = \mathbb{E}|\Delta^m|^2 + c_p\mathbb{E}|g^m - \beta^m|^2\,, \tag{76}$$

where c_p will be carefully chosen later. And the relation between T_1 and T_2 are now in the following lemma:

Lemma 22 *Under conditions of Theorem 10, if $\{x^m\}$ is defined in (31), $\{y^m\}$ is defined in (50) and $\{T^m_1, T^m_2\}$ comes from (76), then for any $a > 0, m \geq 0$, there are upper bounds of T^m_1 and T^m_2:*

$$T^{m+1}_1 \leq (1+a)AT^m_1 + (1+a)BT^m_2 + (1+a)h^3C + \left(1+\frac{1}{a}\right)h^4D\,, \tag{77}$$

and

$$T^{m+1}_2 \leq \widetilde{A}T^m_1 + \widetilde{B}T^m_2\,, \tag{78}$$

where
$$A = 1 - 2\mu h + 3(1+3d)L^2h^2\,, \quad B = 9h^2d\,,$$
$$C = 2L^2d + 72L^2d^3\left[hLd+1\right]\,, \quad D = \frac{1}{2}\left(H^2d^2 + L^3d\right)\,,$$
$$\widetilde{A} = \frac{L^2}{d}\,, \quad \widetilde{B} = 1 - 1/d\,.$$

We refer interested readers to the proof in (Ding and Li, 2020), where we present how to choose proper a and c_p in the lemma above to obtain Theorem 10. Here, we still comment on the dependence of h on d again. Due to the complicated Lyapunov function, and the involvement of $\beta^m - g^m$, the intuition is not as straightforward as it was for Theorem 3 and 9. However, according to Lemma 22, $\widetilde{A}$ is as small as $1/d$ while $\widetilde{B}$ is almost 1, meaning T_2^m barely changes. Moreover, c_p is as small as h^2 and is neglected from the calculation quickly. The dominating terms, in the end, are still the C term and the D term in (77), both of which contribute h^3. The highest power of d in these two terms is the d^3 term that comes from the last term in C, and this is introduced by $\mathbb{E}|E^m|^2$ in (75). This h^3d^3 term, upon iteration, becomes the h^2d^3 term as shown in the theorem, which finally explains the $d^{3/2}$ power in the numerical cost.

As shown in Theorem 3 and 9, in this theorem, we once again encounter the situation where the error of $\mathbb{E}|E^m|^2$ dominates the term in the final iteration formula.

8. Proof of convergence results for U-LMC

In this section, we prove the convergence results for RCD-U-LMC, RCAD-U-LMC, and SVRG-U-LMC.

8.1 Iteration formula for U-LMC

Recall the definition $E^m = \nabla f(x^m) - F^m$, as defined in (20).

According to Algorithm 2, 4, all three algorithms can be seen as drawing (x^0, v^0) from distribution induced by q_0^U, and update (x^m, v^m) using the following coupled SDEs:

$$\begin{cases} \mathrm{V}_t = v^m e^{-2h} - \gamma \int_{mh}^t e^{-2(t-s)}\,\mathrm{d}s F^m + \sqrt{4\gamma}\int_{mh}^t e^{-2(t-s)} dB_s \\ \mathrm{X}_t = x^m + \int_{mh}^t \mathrm{V}_s ds \end{cases}, \tag{79}$$

where B_s is the Brownian motion and $(x^{m+1}, v^{m+1}) = (\mathrm{X}_{(m+1)h}, \mathrm{V}_{(m+1)h})$.

We then define $w^m = x^m + v^m$, and denote $u_m(x,w)$ the distribution of (x^m, w^m) and $u^*(x,w)$ the distribution of (x,w) if $(x, v = w - x)$ is distributed according to density function p_2. From (Cheng et al., 2018, Lemma 8), we have:

$$|x^m - x|^2 + |v^m - v|^2 \le 4(|x^m - x|^2 + |w^m - w|^2) \le 16(|x^m - x|^2 + |v^m - v|^2) \tag{80}$$

and
$$W_2^2(q_m^U, p_2) \le 4W_2^2(u_m, u^*) \le 16W_2^2(q_m^U, p_2)\,. \tag{81}$$

Similar to O-LMC, define another trajectory of sampling by setting $(\widetilde{x}^0, \widetilde{v}^0)$ to be drawn from distribution induced by p_2, and let $\widetilde{x}^m = \widetilde{\mathrm{X}}_{mh}$, $v^m = \widetilde{\mathrm{V}}_{mh}$ be samples from $\left(\widetilde{\mathrm{X}}_t, \widetilde{\mathrm{V}}_t\right)$ that satisfy

$$\begin{cases} \widetilde{\mathrm{V}}_t = \widetilde{v}^0 e^{-2t} - \gamma \int_0^t e^{-2(t-s)} \nabla f\left(\widetilde{\mathrm{X}}_s\right) \mathrm{d}s + \sqrt{4\gamma} e^{-2t} \int_0^t e^{2s} dB_s \\ \widetilde{\mathrm{X}}_t = \widetilde{x}^0 + \int_0^t \widetilde{\mathrm{V}}_s ds \end{cases} \tag{82}$$

with the same Brownian motion as before.

Clearly $\left(\widetilde{\mathrm{X}}_t, \widetilde{\mathrm{V}}_t\right)$ can be seen as drawn from distribution induced by p_2 for all t, and initially we can pick $(\widetilde{x}^0, \widetilde{v}^0)$ such that

$$W_2^2(u_0, u^*) = \mathbb{E}\left(|x^0 - \widetilde{x}^0|^2 + |w^0 - \widetilde{w}^0|^2\right) .$$

We then also define the error Δ^m:

$$\Delta^m = \sqrt{|\widetilde{x}^m - x^m|^2 + |\widetilde{w}^m - w^m|^2} . \tag{83}$$

Similar to Theorem 15, we also have an important iteration formula for $|\Delta^m|^2$:

Theorem 23 *Assume f satisfies Assumption 3.1, let $\{(x^m, v^m)\}$ defined in* (79)*, $\{(\widetilde{x}^m, \widetilde{v}^m)\}$ defined in* (82) *and $\{\Delta^m\}$ defined in* (83)*, then there exists a uniform constant $D > 0$ such that if h satisfies*

$$h < \frac{1}{100(1+D)\kappa} ,$$

then for any $m \geq 0$, we have

$$\mathbb{E}|\Delta^{m+1}|^2 \leq A\mathbb{E}|\Delta^m|^2 + B\mathbb{E}|E^m|^2 + C , \tag{84}$$

where

$$A = 1 - h/(2\kappa) + 480\kappa h^3, \quad B = 10\gamma^2 h^2 , \quad C = 30h^3 d/\mu .$$

Proof

We first divide $|\Delta^{m+1}|^2$ into different parts , and compare (79) and (82) for:

$$\begin{aligned} |\Delta^{m+1}|^2 = & \Bigg| (\widetilde{v}^m - v^m)e^{-2h} + (\widetilde{x}^m - x^m) + \int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s \,\mathrm{d}s \\ & - \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} \left[\nabla f\left(\widetilde{\mathrm{X}}_s\right) - \nabla f(x^m)\right] \mathrm{d}s \\ & + \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} E^m \,\mathrm{d}s \Bigg|^2 \\ & + \left| (\widetilde{x}^m - x^m) + \int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s \,\mathrm{d}s \right|^2 \\ = & |\mathrm{J}_1^m|^2 + |\mathrm{J}_2^m|^2 = \left|\mathrm{J}_1^{r,m} + \mathrm{J}_1^{E,m}\right|^2 + |\mathrm{J}_2^m|^2 , \end{aligned}$$

where we denote

$$\begin{aligned}\mathrm{J}_1^{r,m} = (\widetilde{v}^m - v^m)e^{-2h} + (\widetilde{x}^m - x^m) + \int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s \,\mathrm{d}s \\ - \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} \left[\nabla f\left(\widetilde{\mathrm{X}}_s\right) - \nabla f(x^m)\right] \mathrm{d}s\end{aligned} \tag{85}$$

and

$$\mathrm{J}_1^{E,m} = \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} E^m \,\mathrm{d}s\,, \quad \mathrm{J}_2^m = (\widetilde{x}^m - x^m) + \int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s \,\mathrm{d}s\,. \tag{86}$$

To control J_1^m, we realize all terms in $\mathrm{J}_1^{r,m}$, except V_s, are independent of r^m, and thus,

$$\mathbb{E}\langle \mathrm{J}_1^{r,m}, \mathrm{J}_1^{E,m}\rangle = -\mathbb{E}\left(\mathbb{E}_{r^m}\left\langle \int_{mh}^{(m+1)h} \mathrm{V}_s \,\mathrm{d}s, \mathrm{J}_1^{E,m}\right\rangle\right)\,, \tag{87}$$

where all other terms are eliminated during the process of taking $\mathbb{E}_{r^m}$. Therefore, we can bound $\mathbb{E}\,|\mathrm{J}_1^m|^2$ by

$$\begin{aligned}\mathbb{E}\,|\mathrm{J}_1^m|^2 &= \mathbb{E}\left|\mathrm{J}_1^{r,m} + \mathrm{J}_1^{E,m}\right|^2 \\ &=\mathbb{E}\,|\mathrm{J}_1^{r,m}|^2 + \mathbb{E}\left|\mathrm{J}_1^{E,m}\right|^2 + 2\mathbb{E}\left\langle \mathrm{J}_1^{r,m}, \mathrm{J}_1^{E,m}\right\rangle \\ &\le\mathbb{E}\,|\mathrm{J}_1^{r,m}|^2 + \gamma^2 h^2 \mathbb{E}\,|E^m|^2 + 2\gamma^2 h^3 \mathbb{E}|E^m|^2\,,\end{aligned}$$

where we used

$$\mathbb{E}\left|\mathrm{J}_1^{E,m}\right|^2 \le \gamma^2 h^2 \mathbb{E}\,|E^m|^2$$

and

$$\begin{aligned}&2\mathbb{E}\langle \mathrm{J}_1^{r,m}, \mathrm{J}_1^{E,m}\rangle \\ =&-2\mathbb{E}\left\langle \int_{mh}^{(m+1)h} \mathrm{V}_s \,\mathrm{d}s, \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} \,\mathrm{d}s E^m\right\rangle \\ =&2\mathbb{E}\left\langle \gamma \int_{mh}^{(m+1)h} \int_{mh}^{s} e^{-2(s-t)} \,\mathrm{d}t \,\mathrm{d}s E^m, \gamma \int_{mh}^{(m+1)h} e^{-2((m+1)h-s)} \,\mathrm{d}s E^m\right\rangle, \\ \le&2\gamma^2 h^3 \mathbb{E}|E^m|^2\end{aligned}$$

In conclusion, we have

$$\mathbb{E}\left|\Delta^{m+1}\right|^2 \le\mathbb{E}\,|\mathrm{J}_1^{r,m}|^2 + |\mathrm{J}_2^m|^2 + \left(\gamma^2 h^2 \mathbb{E}|E^m|^2 + 2\gamma^2 h^3 \mathbb{E}|E^m|^2\right)\,, \tag{88}$$

which implies

$$\mathbb{E}\left|\Delta^{m+1}\right|^2 \le\mathbb{E}\,|\mathrm{J}_1^{r,m}|^2 + |\mathrm{J}_2^m|^2 + 2\gamma^2(h^2 + 3h^3)\mathbb{E}|E^m|^2\,. \tag{89}$$

According to Proposition 29 (114), for any $a > 0$, we have

$$\mathbb{E}\,|\mathrm{J}_1^{r,m}|^2 + |\mathrm{J}_2^m|^2 \le C_1 \mathbb{E}|\Delta^m|^2 + 5(1 + 1/a)\gamma^2 h^4 \mathbb{E}(|E^m|^2) + 5(1 + 1/a)\gamma h^4 d\,,$$

where

$$C_1 = (1+a)[1-h/\kappa+200h^2]+80(1+1/a)h^4\,.$$

Plug into (89), we have

$$\mathbb{E}|\Delta^m|^2 \leq C_1\mathbb{E}|\Delta^m|^2+C_2\mathbb{E}|E^m|^2+C_3\,, \tag{90}$$

where we use $\gamma L=1, h<1$ for:

$$C_2=\gamma^2\left[5(1+1/a)h^4+8h^2\right]\,,\quad C_3=5(1+1/a)\gamma h^4 d\,.$$

Considering $h\leq\frac{1}{600\kappa}$, we have

$$1-h/\kappa+200h^2\leq 1-2h/(3\kappa)\,,$$

and thus by setting a so that

$$1+a=\frac{1-h/(2\kappa)}{1-2h/(3\kappa)}\,,\quad \text{which leads}\quad 1+1/a\leq 6\kappa/h\,,$$

we have

$$C_1\leq 1-h/(2\kappa)+480\kappa h^3\,,\quad C_2\leq 30\gamma^2\kappa h^3+8\gamma^2h^2\,,\quad C_3\leq 30\kappa\gamma h^3 d\,.$$

Noting $h<\frac{1}{600\kappa}$, we have $30\kappa h^3<2h^2$, making:

$$C_1\leq A\,,\quad C_2\leq B=10\gamma^2h^2\,,\quad C_3\leq C=30h^3d/\mu\,,$$

finishing the proof. ■

Similar to before we need to make a list of comments:

Remark 24 *Several comments are in order.*

- *The proof is completely independent of $\mathbb{E}|E^m|^2$, meaning the lemma holds true for all U-LMC. The differences between different sampling methods come in through the $\mathbb{E}|E^m|^2$ term that will be analyzed in the following subsections respectively.*
- *Similar to O-LMC, the last term of* (84) *corresponds to the discretization error of U-LMC. In the ideal case, let us suppose $\mathbb{E}|E^m|^2$ is negligible, then besides the iteration term (A term), the remainder term C is at the order of h^3d. Upon iterations, this will finally lead to a formula similar to $W_m<\exp(-hm)W_0+hd^{1/2}$. With the same argument as before, we have $h<\frac{\epsilon}{d^{1/2}}$, and $m>1/h$ for an ϵ accuracy. This is the optimal result one could hope for in the underdamped LMC case. As will be shown below, the $\mathbb{E}|E^m|^2$ is a dominating term for RCD-U-LMC, and such optimal rate is not achieved. However, in the other two cases, $\mathbb{E}|E^m|^2$ is not a dominating term if ϵ is small enough, this means a rate the same as U-LMC is possible for small ϵ.*

8.2 Proof of Theorem 6

To prove Theorem 6, we note that the definition of E is the same as in (20), and the results in Lemma 17 can be recycled:

Proof [Proof of Theorem 6] Recall Lemma 17, we have:

$$\mathbb{E}(|E^m|^2) < 2dL^2\mathbb{E}|\Delta^m|^2 + 2L^2d^2/\mu\,. \tag{91}$$

Plug (91) into (84), we have

$$\begin{aligned}\mathbb{E}|\Delta^{m+1}|^2 &\leq (A + 2dL^2B)\mathbb{E}|\Delta^m|^2 + 2L^2d^2B/\mu + C\\ &\leq C_1\mathbb{E}|\Delta^m|^2 + 2L^2d^2B/\mu + C\,.\end{aligned} \tag{92}$$

Recalling

$$A = 1 - h/(2\kappa) + 480\kappa h^3, \quad B = 10\gamma^2h^2\,, \quad C = 30h^3d/\mu\,,$$

we have, due to $\gamma L = 1$:

$$C_1 = 1 - h/(2\kappa) + 480\kappa h^3 + 20dh^2\,.$$

For h small enough $\left(h < \frac{1}{880d\kappa}\right)$, it is clear that

$$C_1 < 1 - h/(4\kappa)\,. \tag{93}$$

Plug (93) into (92), we have

$$\mathbb{E}|\Delta^{m+1}|^2 \leq (1 - h/(4\kappa))\,\mathbb{E}|\Delta^m|^2 + 20d^2h^2/\mu + 30h^3d/\mu\,, \tag{94}$$

which implies

$$\begin{aligned}\mathbb{E}|\Delta^m|^2 &\leq \exp\left(-hm/(4\kappa)\right)\mathbb{E}|\Delta^0|^2 + 80\kappa d^2h/\mu + 120\kappa h^2d/\mu\\ &\leq \exp\left(-hm/(4\kappa)\right)\mathbb{E}|\Delta^0|^2 + 100\kappa d^2h/\mu\end{aligned}$$

after iteration. Taking square root on each term and using (81), we prove (23). ■

Remark 25 *The result comes with no surprise. The discretization error term C contributes $O(h^3d)$ but the newly induced $\mathbb{E}|E|^2$ term from* (62) *contributes $O(h^2d^2)$. This h^2d^2 term, upon iteration, drops down to hd^2, which explains the d^2 dependence in the end.*

8.3 Proof of Theorem 12

We first have the lemma bounding $\mathbb{E}|E^m|^2$:

Lemma 26 *Assume f satisfies Assumption 3.1, and let $\{(x^m, v^m)\}$ be defined in algorithm 3 (underdamped), $\{(\widetilde{x}^m, \widetilde{v}^m)\}$ be defined in* (82)*, then for any $m \geq 0$, call*

$$s = \left\lfloor \frac{m}{\tau} \right\rfloor\,,$$

we have

$$\mathbb{E}\,|E^m|^2 \leq 3dL^2\left[\mathbb{E}\,|\Delta^m|^2 + \mathbb{E}\,|\Delta^{s\tau}|^2 + 2\tau^2h^2\gamma d\right]\,. \tag{95}$$

We put the proof in Appendix C. Now, we are ready to prove the Theorem 12.

Proof [Proof of Theorem 12] Use Theorem 23 (84), we first bound $\mathbb{E}|E^m|^2$ by plugging in Lemma 26 equation (95),

$$\mathbb{E}|\Delta^{m+1}|^2 \leq (A+3dL^2B)\mathbb{E}|\Delta^m|^2 + 3dL^2B\mathbb{E}|\Delta^{s\tau}|^2 + 6Ld^2\tau^2h^2B + C\,.$$

Recalling $A = 1 - h/(2\kappa) + 480\kappa h^3$, $B = 10\gamma^2h^2$ and $C = 30h^3d/\mu$, we have

$$A + 3dL^2B \leq 1 - h/(4\kappa)\,, \quad \text{when} \quad h < \frac{1}{1648d\kappa}$$

and thus

$$\mathbb{E}|\Delta^{m+1}|^2 \leq \left(1-\frac{h}{4\kappa}\right)\mathbb{E}|\Delta^m|^2 + 30h^2d\mathbb{E}|\Delta^{s\tau}|^2 + 30h^3d/\mu + 60\gamma h^4\tau^2d^2\,. \tag{96}$$

Use $\gamma = \frac{1}{L} = \frac{1}{\kappa\mu}$, we have:

$$\mathbb{E}|\Delta^m|^2 < \left(1-\frac{h}{4\kappa}\right)^{m-s\tau}\mathbb{E}|\Delta^{s\tau}|^2 + 30h^2(m-s\tau)d\mathbb{E}|\Delta^{s\tau}|^2 + 30h^3\tau d/\mu + 60h^4\tau^2d^2/(\kappa\mu)\,. \tag{97}$$

Similar to (71), by setting $m = (s+1)\tau$, we also have:

$$\begin{aligned}\mathbb{E}|\Delta^{(s+1)\tau}|^2 &< \left(1-\frac{h\tau}{8\kappa}\right)\mathbb{E}|\Delta^{s\tau}|^2 + 30h^2\tau d\mathbb{E}|\Delta^{s\tau}|^2 + 30h^3\tau d/\mu + 60h^4\tau^3d^2/(\kappa\mu)\\ &< \left(1-\frac{h\tau}{16\kappa}\right)\mathbb{E}|\Delta^{s\tau}|^2 + 30h^3\tau d/\mu + 60h^4\tau^3d^2/(\kappa\mu)\,,\end{aligned} \tag{98}$$

where we absorb the $30h^2\tau d$ term into $\frac{h\tau}{16\kappa}$ for small enough h (for example when $h < \frac{1}{1648d\kappa}$) in the second inequality. This finally gives:

$$\mathbb{E}|\Delta^{s\tau}|^2 < \left(1-\frac{h\tau}{16\kappa}\right)^s\mathbb{E}|\Delta^0|^2 + 480h^2d\kappa/\mu + 960h^3\tau^2d^2/\mu\,. \tag{99}$$

Plug (99) back into (97), we have:

$$\begin{aligned}\mathbb{E}|\Delta^m|^2 &< \left(\left(1-\frac{h}{4\kappa}\right)^{m-s\tau} + 30h^2(m-s\tau)d\right)\left(1-\frac{h\tau}{16\kappa}\right)^s\mathbb{E}|\Delta^0|^2 + 510h^2d\kappa/\mu + 1020h^3\tau^2d^2/\mu\\ &< \left(1-\frac{h}{16\kappa}\right)^m\mathbb{E}|\Delta^0|^2 + 510h^2d\kappa/\mu + 1020h^3\tau^2d^2/\mu\\ &< \exp\left(-\frac{hm}{16\kappa}\right)\mathbb{E}|\Delta^0|^2 + 510h^2d\kappa/\mu + 1020h^3\tau^2d^2/\mu\,,\end{aligned}$$

where in the first inequality we used $h\tau < 1 \leq \kappa$, and in the second inequality we used $h(m-s\tau) \leq h\tau < \frac{1}{40}$ and $30h^2(m-s\tau)d < 1 - \frac{(m-s\tau)h}{16\kappa}$ to obtain

$$\begin{aligned}\left(1-\frac{h}{4\kappa}\right)^{m-s\tau} + 30h^2(m-s\tau)d < 1 - \frac{(m-s\tau)h}{8\kappa} + 30h^2(m-s\tau)d < 1 - \frac{(m-s\tau)h}{16\kappa}\\ \leq \left(1-\frac{h}{16\kappa}\right)^{m-s\tau}\,.\end{aligned}$$

Taking square root on both sides gives the conclusion (37). ■

Again, we discuss the newly introduced error by $\mathbb{E}|E^m|^2$. Take $\tau = d$ in Lemma 26, this part of error is at the order of h^4d^4 when entering the iteration formula with an extra h^2 term from B. By induction, one h drops out and upon taking square root, this part of error is at the order of $h^{3/2}d^2$. Making it smaller than ϵ gives $h < \frac{\epsilon^{2/3}}{d^{4/3}}$. When ϵ is small enough, this is a more relaxed constraint than the original discretization constraint $h < \frac{\epsilon^{1/2}}{d}$.

8.4 Sketch proof of Theorem 13

Similar to section 7.4, we have the following estimation for $\mathbb{E}\,|E^m|^2$:

Lemma 27 *Assume f satisfies Assumption 3.1, if $\{(x^m, v^m)\}$ is defined in algorithm 4 (underdamped), $\{(\widetilde{x}^m, \widetilde{v}^m)\}$ is defined in* (82)*, then for any $m \geq 0$, we have*

$$\mathbb{E}(|E^m|^2) \leq 3dL^2\mathbb{E}|\Delta^m|^2 + 24Lh^2d^4 + 3d\mathbb{E}\,|\beta^m - g^m|^2\ . \tag{100}$$

We leave out the proof and refer interested readers to (Ding and Li, 2020). Here, β^m and g are defined same as (74) and (48) (y change to $\widetilde{x}$) and we will show the decay of the following Lyapunov function:

$$T^m \triangleq T_1^m + c_pT_2^m = \mathbb{E}\left(|\widetilde{x}^m - x^m|^2 + |\widetilde{w}^m - w^m|^2\right) + c_p\mathbb{E}|g^m - \beta^m|^2\,, \tag{101}$$

where c_p will be carefully chosen later. And the relation between T_1 and T_2 are now in the following lemma:

Lemma 28 *Under conditions of Theorem 13, if $\{(x^m, v^m)\}$ is defined in algorithm 4 (underdamped), $\{(\widetilde{x}^m, \widetilde{v}^m)\}$ is defined in* (82) *and $\{(T_1^m, T_2^m)\}$ comes from* (101)*, then for any $m \geq 0$, we have*

$$T_1^{m+1} < D_1T_1^m + D_2T_2^m + D_3\,, \tag{102}$$

$$T_2^{m+1} \leq \frac{L^2}{d}T_1^m + \left(1 - \frac{1}{d}\right)T_2^m\,, \tag{103}$$

where

$$D_1 = 1 - h/(2\kappa) + 40h^2d,\ D_2 = 30\gamma^2h^2d,\ D_3 = 240\gamma h^4d^4 + 30h^3d/\mu\,.$$

The proof is found in (Ding and Li, 2020). As in the case of an overdamped algorithm, one needs to tune a and c_p in the lemma above to obtain Theorem 13. The result here is also similar to the previous ones. T_2^m term enters in the Lyapunov function with an h^2 rate and is negligible and the iteration formula is dominated by the D_3 term in (102). There are two main terms here, the h^4d^4 mainly comes from $\mathbb{E}|E^m|^2$ while the h^3d is from the original iteration formula (84). It is hard to compare which term dominates. Indeed with smaller ϵ, the requirement from the h^3d term is more restrictive. This gives the last two terms in (39).

Acknowledgments

Q.L. acknowledges support from Vilas Early Career award. The research is supported in part by NSF via grant DMS-1750488, TRIPODS 1740707 and 2023239 and Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin Madison with funding from the Wisconsin Alumni Research Foundation. Both authors appreciate valuable suggestions from the two anonymous reviewers and discussions with Stephen J. Wright.

Appendix A. Proof of Lemma 17

We prove (62) in this section. Recall the definition of E^m to be

$$E^m = \nabla f(x^m) - F^m\,.$$

To bound variance of E^m, we consider each component

$$\mathbb{E}(|E^m|^2) = \mathbb{E}\left(\mathbb{E}_r(|\partial_i f(x^m) - d\partial_r f(x^m)\mathbf{e}^r_i|^2)\right) = (d-1)\mathbb{E}|\partial_i f(x^m)|^2\,,$$

where we use r is uniformly chosen from $1, 2, \ldots, d$, this implies

$$\mathbb{E}(|E^m_i|^2) = \sum_{i=1}^{d} \mathbb{E}(|E^m_i|^2) < \mathbb{E}|\nabla f(x^m)|^2 d\,. \tag{104}$$

Since gradient of f is L-Lipschitz, we have

$$\begin{aligned} |\nabla f(x^m)|^2 &\le 2|\nabla f(x^m) - \nabla f(y^m)|^2 + 2|\nabla f(y^m)|^2 \\ &\le 2L^2|x^m - y^m|^2 + 2|\nabla f(y^m)|^2\,, \end{aligned} \tag{105}$$

and

$$\mathbb{E}|\nabla f(y^m)|^2 \le dL\,, \tag{106}$$

where we use $y_t \sim p$ for any t and Lemma 3 in (Dalalyan and Karagulyan, 2019) in the last inequality. Plug (105),(106) into (104), we have

$$\mathbb{E}(|E^m|^2) < 2dL^2\mathbb{E}|\Delta^m|^2 + 2d^2L\,, \tag{107}$$

which proves (62).

Appendix B. Proof of Lemma 20

Let

$$\widetilde{x} = x^{s\tau}, \quad \widetilde{y} = y^{s\tau}$$

For $m = s\tau$, according to (27),

$$\mathbb{E}\,|E^m|^2 = (d-1)\mathbb{E}\,|\nabla f(x^m) - \nabla f(\widetilde{x})|^2 = 0\,.$$

Therefore, we only need to consider $m > s\tau$, according to (27), we directly have

$$\mathbb{E}\,|E^m|^2 = (d-1)\mathbb{E}\,|\nabla f(x^m) - \nabla f(\widetilde{x})|^2 \le dL^2\mathbb{E}\,|x^m - \widetilde{x}|^2\,, \tag{108}$$

where the last inequality comes from Hölder's inequality. Then, use Hölder's inequality again, we have

$$\mathbb{E}\,|x^m - \widetilde{x}|^2 \le 3\mathbb{E}\,|x^m - y^m|^2 + 3\mathbb{E}\,|y^m - \widetilde{y}|^2 + 3\mathbb{E}\,|\widetilde{y} - \widetilde{x}|^2\,, \tag{109}$$

The second term can be further bounded by (50)

$$\mathbb{E}\,|y^m - \widetilde{y}|^2 = \mathbb{E}\left|\int_{s\tau h}^{mh} \nabla f(y_s)\,\mathrm{d}s - \sqrt{2h}\sum_{j=s\tau}^{m-1}\xi^j\right|^2 \le 2h^2\tau^2\mathbb{E}_p|\nabla f(y)|^2 + 4h\tau d\,, \tag{110}$$

where we use Hölder's inequality and $y_t \sim p$ for any t in the last inequality. Finally, according to Lemma 3 in (Dalalyan and Karagulyan, 2019) to obtain

$$\mathbb{E}_p|\nabla f(y)|^2 \leq Ld$$

Plug this into (110), combine (108),(109), we obtain (67).

Appendix C. Proof of Lemma 26

The proof is similar to Appendix B

For $m = s\tau$, according to (27),

$$\mathbb{E}\,|E^m|^2 = (d-1)\mathbb{E}\,|\nabla f(x^m) - \nabla f(x^{s\tau})|^2 = 0\,.$$

Therefore, we only need to consider $m > s\tau$, according to (27), we directly have

$$\mathbb{E}\,|E^m|^2 = (d-1)\mathbb{E}\,|\nabla f(x^m) - \nabla f(x^{s\tau})|^2 \leq dL^2\mathbb{E}\,|x^m - x^{s\tau}|^2\ , \tag{111}$$

where the last inequality comes from Hölder's inequality. Then, use Hölder's inequality again, we have

$$\mathbb{E}\,|x^m - x^{s\tau}|^2 \leq 3\mathbb{E}\,|x^m - \widetilde{x}^m|^2 + 3\mathbb{E}\,|\widetilde{x}^m - \widetilde{x}^{s\tau}|^2 + 3\mathbb{E}\,|\widetilde{x}^{s\tau} - x^{s\tau}|^2\ , \tag{112}$$

The second term can be further bounded by (82)

$$\mathbb{E}\,|\widetilde{x}^m - \widetilde{x}^{s\tau}|^2 = \mathbb{E}\left|\int_{s\tau h}^{mh} \widetilde{\mathrm{V}}_s\,\mathrm{d}s\right|^2 \leq 2h^2\tau^2\mathbb{E}_{p_2}|\widetilde{\mathrm{V}}|^2\,, \tag{113}$$

where we use Hölder's inequality and $\widetilde{\mathrm{V}}_t \sim p$ for any t in the last inequality. Finally, since we have

$$\mathbb{E}_p|\widetilde{\mathrm{V}}|^2 = \gamma d\,,$$

plug this into (110), combine (108),(109), we obtain (95).

Appendix D. Key lemmas in the proof of Section 8

In this section, we always assume $h < 1$ and f satisfies assumption 3.1, and all notations come from Section 8.

Proposition 29 $\mathrm{J}^{r,m}_1, \mathrm{J}^m_2$ *are defined in* (85) *and* (86)*, then for any* $a > 0$*, we have*

$$\mathbb{E}\,|\mathrm{J}^{r,m}_1|^2 + |\mathrm{J}^m_2|^2 \leq C_1\mathbb{E}|\Delta^m|^2 + 5(1+1/a)\gamma^2h^4\mathbb{E}(|E^m|^2) + 5(1+1/a)\gamma h^4 d\,, \tag{114}$$

where

$$C_1 = (1+a)[1 - h/\kappa + 200h^2] + 80(1+1/a)h^4\,.$$

We first introduce four new terms to prove Proposition 29. Denote

$$\begin{aligned} A^m =& (\widetilde{v}^m - v^m)(h + e^{-2h}) + (\widetilde{x}^m - x^m) \\ & - \gamma\int_{mh}^{(m+1)h} e^{-2((m+1)h-s)}\left[\nabla f(\widetilde{x}^m) - \nabla f(x^m)\right]\mathrm{d}s\,, \end{aligned} \tag{115}$$

$$\begin{aligned} B^m = &\int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s - (\widetilde{v}^m - v^m)\,\mathrm{d}s \\ &- \gamma\int_{mh}^{(m+1)h} e^{-2((m+1)h-s)}\left[\nabla f\left(\widetilde{\mathrm{X}}_s\right) - \nabla f(\widetilde{x}^m)\right]\,\mathrm{d}s \end{aligned}, \tag{116}$$

$$C^m = (\widetilde{x}^m - x^m) + \int_{mh}^{(m+1)h} \widetilde{v}^m - v^m\,\mathrm{d}s = (\widetilde{x}^m - x^m) + h(\widetilde{v}^m - v^m)\,, \tag{117}$$

$$D^m = \int_{mh}^{(m+1)h} \widetilde{\mathrm{V}}_s - \mathrm{V}_s - (\widetilde{v}^m - v^m)\,\mathrm{d}s\,, \tag{118}$$

Then, we have the following three lemmas:

Lemma 30 *For any $m \geq 0$*

$$\mathbb{E}\int_{mh}^{(m+1)h} \left|\widetilde{\mathrm{X}}_t - \widetilde{x}^m\right|^2\,\mathrm{d}t \leq \frac{h^3\gamma d}{3} \tag{119}$$

and

$$\mathbb{E}\int_{mh}^{(m+1)h} \left|\left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m)\right|^2\,\mathrm{d}t \leq 16h^3\mathbb{E}|\Delta^m|^2 + \gamma^2h^3\mathbb{E}(|E^m|^2) + 0.4\gamma h^5 d\,, \tag{120}$$

Lemma 31 *B^m, D^m are defined in* (116),(118)*, we have for any $m \geq 0$*

$$\mathbb{E}|B^m|^2 \leq 32h^4\mathbb{E}|\Delta^m|^2 + 2\gamma^2h^4\mathbb{E}(|E^m|^2) + 2\gamma h^4 d \tag{121}$$

$$\mathbb{E}|D^m|^2 \leq 16h^4\mathbb{E}|\Delta^m|^2 + \gamma^2h^4\mathbb{E}(|E^m|^2) + 0.4\gamma h^6 d \tag{122}$$

Lemma 32 *A^m, C^m defined in* (115),(117)*, there exists a uniform constant D such that for any $m \geq 0$*

$$\mathbb{E}(|A^m|^2 + |C^m|^2) \leq \left[1 - h/\kappa + 200h^2\right]\mathbb{E}|\Delta^m|^2 \tag{123}$$

where $\kappa = L/\mu$ is the condition number of f.

Now, we are ready to prove Proposition 29:

Proof [Proof of Proposition 29] First, we notice that

$$\mathrm{J}_1^{r,m} = A^m + B^m, \quad \mathrm{J}_2^m = C^m + D^m\,.$$

By Young's inequality, we have

$$\begin{aligned} \mathbb{E}|\mathrm{J}_1^{r,m}|^2 + \mathbb{E}|\mathrm{J}_2^m|^2 =& \mathbb{E}|A^m + B^m|^2 + \mathbb{E}|C^m + D^m|^2 \\ \leq& (1+a)\left(\mathbb{E}|A^m|^2 + \mathbb{E}|C^m|^2\right) \\ &+ (1+1/a)(\mathbb{E}|B^m|^2 + \mathbb{E}|D^m|^2)\,, \end{aligned} \tag{124}$$

where $a > 0$ will be carefully chosen later. Now, the first term of (124) only contains information from previous step, using f is strongly convex, we can bound it using $|\Delta^m|^2$ (shown in Appendix D Lemma 32). To bound the second term, we need to consider difference between x, v at t_{m+1} and t_m, which can be bounded by $|\Delta^m|^2$ and $|E^m|^2$ (shown in Appendix D Lemma 31).

According to Lemma 31-32, we first have

$$\begin{aligned}
&\mathbb{E}|\mathrm{J}_1^{r,m}|^2 + \mathbb{E}|\mathrm{J}_2^m|^2 \\
\le&(1+a)\left[1 - h/\kappa + 200h^2\right]\mathbb{E}|\Delta^m|^2 \\
&+ (1+1/a)\left[80h^4\mathbb{E}|\Delta^m|^2 + 5\gamma^2h^4\mathbb{E}(|E^m|^2) + 5\gamma h^4 d\right] \\
=&C_1\mathbb{E}|\Delta^m|^2 + 5(1+1/a)\gamma^2h^4\mathbb{E}(|E^m|^2) + 5(1+1/a)\gamma h^4 d\,,
\end{aligned}$$

where in the first inequality we use $1 + h^2 < 2$ and

$$C_1 = (1+a)[1 - h/\kappa + 200h^2] + 80(1+1/a)h^4\,.$$

■

To complete the proof, we prove three Lemmas one by one:

Proof [Proof of Lemma 30] First we prove (119). According to (82), we have

$$\begin{aligned}
\mathbb{E}\int_{mh}^{(m+1)h}\left|\widetilde{\mathrm{X}}_t - \widetilde{x}^m\right|^2 dt &= \mathbb{E}\int_{mh}^{(m+1)h}\left|\int_{mh}^t \widetilde{\mathrm{V}}_s ds\right|^2 dt \\
&\le \int_{mh}^{(m+1)h}(t-mh)\int_{mh}^t \mathbb{E}\left|\widetilde{\mathrm{V}}_s\right|^2 dsdt \qquad (125)\\
&= \int |v|^2 p_2(x,v)\,\mathrm{d}x\,\mathrm{d}v\int_{mh}^{(m+1)h}(t-mh)^2dt = \frac{h^3\gamma d}{3}\,,
\end{aligned}$$

where in the first inequality we use Hölder's inequality, and for the second equality we use p_2 is a stationary distribution so that $\left(\widetilde{\mathrm{X}}_t, \widetilde{\mathrm{V}}_t\right) \sim p_2$ and $\widetilde{\mathrm{V}}_t \sim \exp(-|v|^2/(2\gamma))$ for any t.

Second, to prove (120), using (79),(82), we first rewrite $\left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m)$ as

$$\begin{aligned}
\left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m) =& (\widetilde{v}^m - v^m)\left(e^{-2(t-mh)} - 1\right) \\
&- \gamma\int_{mh}^t e^{-2(t-s)}\left[\nabla f(\widetilde{\mathrm{X}}_s) - \nabla f(x^m)\right]\mathrm{d}s \\
&+ \gamma\int_{mh}^t e^{-2(t-s)}\,\mathrm{d}sE^m \qquad (126)\\
=&\mathrm{I}(t) + \mathrm{II}(t) + \mathrm{III}(t)\,.
\end{aligned}$$

for $mh \le t \le (m+1)h$. Then we bound each term separately:

- $$\begin{aligned}\mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{I}(t)|^2 \,\mathrm{d}t &\le h\mathbb{E}\int_{mh}^{(m+1)h} \left|(\widetilde{v}^m - v^m)\left(e^{-2(t-mh)}-1\right)\right|^2 \mathrm{d}t \\ &\le h\int_{mh}^{(m+1)h} (2(t-mh))^2\mathbb{E}\left|\widetilde{v}^m - v^m\right|^2 \mathrm{d}t \\ &\le \frac{4h^3}{3}\mathbb{E}\left|\widetilde{v}^m - v^m\right|^2 ,\end{aligned} \tag{127}$$

  where we use Hölder's inequality in the first inequality and $1-e^{-x} < x$ in the second inequality.

- $$\begin{aligned}&\mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{II}(t)|^2 \,\mathrm{d}t \le \gamma^2\mathbb{E}\int_{mh}^{(m+1)h} \left|\int_{mh}^{t} e^{-2(t-s)}\left[\nabla f(\widetilde{\mathrm{X}}_s) - \nabla f(x^m)\right] \mathrm{d}s\right|^2 \mathrm{d}t \\ \le& 2\gamma^2\mathbb{E}\int_{mh}^{(m+1)h} \left|\int_{mh}^{t} e^{-2(t-s)}\left[\nabla f(\widetilde{\mathrm{X}}_s) - \nabla f(\widetilde{x}^m)\right] \mathrm{d}s\right|^2 \mathrm{d}t \\ &+ 2\gamma^2\mathbb{E}\int_{mh}^{(m+1)h} \left|\int_{mh}^{t} e^{-2(t-s)}\left[\nabla f(\widetilde{x}^m) - \nabla f(x^m)\right] \mathrm{d}s\right|^2 \mathrm{d}t \\ \le& 2\gamma^2\int_{mh}^{(m+1)h} (t-mh)\mathbb{E}\int_{mh}^{t} \left|\nabla f(\widetilde{\mathrm{X}}_s) - \nabla f(\widetilde{x}^m)\right|^2 \mathrm{d}s\,\mathrm{d}t \\ &+ 2\gamma^2\int_{mh}^{(m+1)h} (t-mh)\mathbb{E}\int_{mh}^{t} \left|\nabla f(\widetilde{x}^m) - \nabla f(x^m)\right|^2 \mathrm{d}s\,\mathrm{d}t \\ \le& 2\gamma^2L^2\int_{mh}^{(m+1)h} (t-mh)\mathbb{E}\int_{mh}^{t} \left|\widetilde{\mathrm{X}}_s - \widetilde{x}^m\right|^2 \mathrm{d}s\,\mathrm{d}t \\ &+ 2\gamma^2L^2\int_{mh}^{(m+1)h} (t-mh)\mathbb{E}\int_{mh}^{t} \left|\widetilde{x}^m - x^m\right|^2 \mathrm{d}s\,\mathrm{d}t \\ \le& 2\gamma^3L^2d\int_{mh}^{(m+1)h} \frac{(t-mh)^4}{3}\,\mathrm{d}t + 2\gamma^2L^2\int_{mh}^{(m+1)h} (t-mh)^2\,\mathrm{d}t\mathbb{E}\left|\widetilde{x}^m - x^m\right|^2 \\ \le& \frac{2\gamma^3L^2h^5d}{15} + \frac{2\gamma^2L^2h^3}{3}\mathbb{E}\left|\widetilde{x}^m - x^m\right|^2 ,\end{aligned} \tag{128}$$

  where in the third inequality we use the gradient of f is L-Lipschitz function and we use (119) in the fourth inequality.

- $$\begin{aligned}\mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{III}(t)|^2 \,\mathrm{d}t &= \gamma^2\mathbb{E}\int_{mh}^{(m+1)h} \left|\int_{mh}^{t} e^{-2(t-s)}\,\mathrm{d}sE^m\right|^2 \mathrm{d}t \\ &\le \gamma^2\int_{mh}^{(m+1)h} (t-mh)^2\,\mathrm{d}t\mathbb{E}(|E^m|^2) \\ &\le \frac{\gamma^2h^3}{3}\mathbb{E}(|E^m|^2),\end{aligned} \tag{129}$$

Plug (127),(128),(129) into (126) and using $\gamma L = 1$, we have

$$\begin{aligned}
&\mathbb{E}\int_{mh}^{(m+1)h} \left|\left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m)\right|^2 \,\mathrm{d}t \\
\leq& 3\left(\mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{I}(t)|^2 \,\mathrm{d}t + \mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{II}(t)|^2 \,\mathrm{d}t + \mathbb{E}\int_{mh}^{(m+1)h} |\mathrm{III}(t)|^2 \,\mathrm{d}t\right) \\
\leq& 4h^3\left(\mathbb{E}\left|\widetilde{x}^m - x^m\right|^2 + \mathbb{E}\left|\widetilde{v}^m - v^m\right|^2\right) + \gamma^2 h^3 \mathbb{E}(|E^m|^2) + 0.4\gamma h^5 d\,,
\end{aligned}$$

using (80), we get the desired result. ■

Proof [Proof of Lemma 31] First, we separate B^m into two parts:

$$\begin{aligned}
\mathbb{E}|B^m|^2 \leq& 2\mathbb{E}\left|\int_{mh}^{(m+1)h} \left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m) \,\mathrm{d}t\right|^2 \\
&+ 2\mathbb{E}\left|\gamma\int_{mh}^{(m+1)h} e^{-2((m+1)h-t)}\left[\nabla f(\widetilde{\mathrm{X}}_t) - \nabla f(\widetilde{x}^m)\right] \,\mathrm{d}t\right|^2 .
\end{aligned}$$

And each terms can be bounded:

- $$\begin{aligned}
&\mathbb{E}\left|\int_{mh}^{(m+1)h} \left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m) \,\mathrm{d}t\right|^2 \\
\leq& h\mathbb{E}\int_{mh}^{(m+1)h} \left|\left(\widetilde{\mathrm{V}}_t - \mathrm{V}_t\right) - (\widetilde{v}^m - v^m)\right|^2 \,\mathrm{d}t \\
\leq& 16h^4\mathbb{E}|\Delta^m|^2 + \gamma^2 h^4 \mathbb{E}(|E^m|^2) + 0.4\gamma h^6 d\,,
\end{aligned} \tag{130}$$

  where we use Lemma 30 (120) in the second inequality.

- $$\begin{aligned}
&\mathbb{E}\left|\gamma\int_{mh}^{(m+1)h} e^{-2((m+1)h-t)}\left[\nabla f(\widetilde{\mathrm{X}}_t) - \nabla f(\widetilde{x}^m)\right] \,\mathrm{d}t\right|^2 \\
\leq& h\gamma^2\mathbb{E}\int_{mh}^{(m+1)h} \left|e^{-2((m+1)h-t)}\left[\nabla f(\widetilde{\mathrm{X}}_t) - \nabla f(\widetilde{x}^m)\right]\right|^2 \,\mathrm{d}t \\
\leq& h\gamma^2 L^2\mathbb{E}\int_{mh}^{(m+1)h} \left|\widetilde{\mathrm{X}}_t - \widetilde{x}^m\right|^2 \,\mathrm{d}t \\
\leq& \frac{h^4\gamma^3 L^2 d}{3} \leq \frac{h^4\gamma d}{3}\,,
\end{aligned} \tag{131}$$

  where we use Lemma 30 (119) and $\gamma L = 1$ in the last two inequalities.

Combine (130),(131) together, we finally have

$$\mathbb{E}|B|^2 \leq 32h^4\mathbb{E}|\Delta^m|^2 + 2\gamma^2 h^4 \mathbb{E}(|E^m|^2) + 0.8h^6\gamma d + 2h^4\gamma d/3\,,$$

which implies (121) if we further use $h<1$.

Next, estimation of $\left(\mathbb{E}|D|^2\right)^{1/2}$ is a direct result of (130). ■

Proof [Proof of Lemma 32] Let $\widetilde{x}^m-x^m=a$ and $\widetilde{w}^m-w^m=b$. First, by the mean-value theorem, there exists a matrix H such that $\mu I_d\preceq H\preceq LI_d$ and

$$\nabla f(\widetilde{x}^m)-\nabla f(x^m)=Ha\,.$$

By calculation, $\int_{mh}^{(m+1)h}e^{-2((m+1)h-t)}\,\mathrm{d}t=\frac{1-e^{-2h}}{2}$ and

$$\begin{aligned}A^m&=(h+e^{-2h})(\widetilde{v}^m-v^m)+\left(I_d-\frac{(1-e^{-2h})}{2}\gamma H\right)(\widetilde{x}^m-x^m)\\&=\left(\left(1-h-e^{-2h}\right)I_d-\frac{(1-e^{-2h})}{2}\gamma H\right)a+(h+e^{-2h})b\end{aligned}\,.$$

$$C^m=(1-h)a+hb\,.$$

Since $\|\gamma H\|_2\le 1$ and we also have the following calculation

$$|1-e^{-2h}-2h|\le 5h^2\,,$$

where we use $h<\frac{1}{1648}$.

If we further define matrix $\mathcal{M}_A$ and $\mathcal{M}_C$ such that

$$|A^m|^2=(a,b)^\top\mathcal{M}_A\,(a,b)\ ,\quad |C^m|^2=(a,b)^\top\mathcal{M}_C\,(a,b)\ ,$$

then, we have

$$\left\|\mathcal{M}_A-\begin{bmatrix}0 & hI_d-\gamma hH\\ hI_d-\gamma hH & (1-2h)I_d\end{bmatrix}\right\|_2\le 100h^2\,,$$

and

$$\left\|\mathcal{M}_B-\begin{bmatrix}(1-2h)I_d & hI_d\\ hI_d & 0\end{bmatrix}\right\|_2\le 100h^2\,,$$

where we use $h<1/1648$ by (38) and $\|\gamma H\|_2\le 1$. This further implies

$$|A^m|^2+|C^m|^2=(a,b)^\top\begin{bmatrix}(1-2h)I_d & 2hI_d-\gamma hH\\ 2hI_d-\gamma hH & (1-2h)I_d\end{bmatrix}(a,b)+h^2\,(a,b)^\top Q\,(a,b)$$

where $\|Q\|_2\le 200$. Calculate the eigenvalue of the dominating matrix (first term), we need to solve

$$\det\left\{(1-2h-\lambda)^2I_d-(2hI_d-\gamma hH)^2\right\}=0\,,$$

which implies eigenvalues $\{\lambda_j\}_{j=1}^d$ solve

$$(1-2h-\lambda_j)^2-(2h-\gamma h\Lambda_j)^2=0\,,$$

where Λ_j is j-th eigenvalue of H. Since $\gamma\Lambda_j\le\gamma L=1$ and $h<1$, we have

$$\lambda_j\le 1-\gamma\Lambda_j h\le 1-\mu h\gamma=1-h/\kappa$$

for each $j = 1, \ldots, d$. This implies

$$\left\| \begin{bmatrix} (1-2h)I_d & 2hI_d - \gamma hH \\ 2hI_d - \gamma hH & (1-2h)I_d \end{bmatrix} \right\|_2 \leq 1 - h/\kappa\,,$$

and

$$|A^m|^2 + |C^m|^2 \leq (1 - h/\kappa + 200h^2)(|a|^2 + |b|^2)\,.$$

Taking expectation on both sides, we obtain (123).

■

References

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

Jack Baker, Paul Fearnhead, Emily Fox, and Christopher Nemeth. Control variates for stochastic gradient MCMC. *Statistics and Computing*, 29, 05 2019.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186, 2010.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, Cambridge, MA, USA, 2006.

Niladri S. Chatterji, Nicolas Flammarion, Yi-An Ma, Peter L. Bartlett, and Michael I. Jordan. On the theory of variance reduction for stochastic gradient Monte Carlo. In *Proceedings of the 35th International Conference on Machine Learning*, pages 764–773, 2018.

Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pages 1683–1691, 2014.

Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, pages 300–323, 2018.

Arnak S. Dalalyan and Avetik Karagulyan. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12): 5278–5311, 2019.

Arnak S. Dalalyan and Lionel Riou-Durand. On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli*, 26(3):1956–1988, 2020.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 1646–1654, 2014.

Zhiyan Ding and Qin Li. Variance reduction for random coordinate descent-Langevin Monte Carlo. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 3748–3760, 2020.

Zhiyan Ding, Qin Li, Jianfeng Lu, and Stephen J. Wright. Random coordinate underdamped langevin monte carlo. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 2701–2709, 2021a.

Zhiyan Ding, Qin Li, Jianfeng Lu, and Stephen J. Wright. Random coordinate langevin monte carlo. In *Proceedings of 34th Conference on Learning Theory*, pages 1683–1710, 2021b.

Simon Duane, A.D. Kennedy, Brian J.Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Kumar Avinava Dubey, Sashank J. Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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

Alain Durmus, Umut Simsekli, Eric Moulines, Roland Badeau, and Gaël RICHARD. Stochastic gradient richardson-romberg Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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

Raaz Dwivedi, Yuansi Chen, M. Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast! *Journal of Machine Learning Research*, 20(183):1–42, 2018.

Andreas Eberle, Arnaud Guillin, and Raphael Zimmer. Couplings and quantitative contraction rates for Langevin dynamics. *Annals of Probability*, 47(4):1982–2010, 2019.

Peter Fabian. Atmospheric sampling. *Advances in Space Research*, 1(11):17–27, 1981.

Alfredo Garbuno-Inigo, Franca Hoffmann, Wuchen Li, and Andrew M. Stuart. Interacting Langevin diffusions: gradient structure and ensemble Kalman sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1):412–441, 2020.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

László Gerencsér. Rate of convergence of moments of Spall's SPSA method. In *1997 European Control Conference (ECC)*, pages 2192–2197, 1997.

John Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, 1989.

Wilfred Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, 2008.

Marco A. Iglesias, Kody J. H. Law, and Andrew M. Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 315–323, 2013.

Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23(3):462–466, 1952.

Nathan L. Kleinman, James C. Spall, and Daniel Q. Naiman. Simulation-based optimization with stochastic approximation using common random numbers. *Management Science*, 45 (11):1570–1578, 1999.

Günter Leugering, Sebastian Engell, Andreas Griewank, Michael Hinze, Rolf Rannacher, Volker Schulz, Michael Ulbrich, and Stefan Ulbrich. *Constrained Optimization and Optimal Control for Partial Differential Equations.* International Series of Numerical Mathematics. Birkhäuser Basel, 2012.

Ruiyn Li, Sen Pei, Bin Chen, Yimeng Song, Tao Zhang, Wan Yang, and Jeffrey Shaman. Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). *Science*, 368(6490):489–493, 2020.

Yi-An Ma, Tianqi Chen, and Emily B. Fox. A complete recipe for stochastic gradient MCMC. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 2917—-2925, 2015.

Peter Markowich and Cédric Villani. On the trend to equilibrium for the Fokker-Planck equation: An interplay between physics and functional analysis. In *Physics and Functional Analysis, Matematica Contemporanea (SBM) 19*, pages 1–29, 1999.

James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar Ghattas. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3A):1460–1487, 2012.

Benedict C. May, Nathan Korda, Anthony Lee, and David S. Leslie. Optimistic Bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(1): 2069—-2106, 2012.

Eric Mazumdar, Aldo Pacchiano, Yi-An Ma, Peter L. Bartlett, and Michael I. Jordan. On approximate Thompson sampling with Langevin algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6797–6807, 2020.

Kerrie Mengersen and Richard L. Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):101–121, 1996.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

Wenlong Mou, Yi-An Ma, Martin J. Wainwright, Peter L. Bartlett, and Michael I. Jordan. High-order Langevin diffusion yields an accelerated MCMC algorithm. *Journal of Machine Learning Research*, 22(42):1–41, 2021.

Lawrence M. Murray, Anthony Lee, and Pierre E. Jacob. Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3):789–805, 2016.

Narayana Rao Nagarajan, Mehdi Matt Honarpour, and Krishnaswamy Sampath. Reservoir-fluid sampling and characterization — key to efficient reservoir management. *Journal of Petroleum Technology*, 59(8):80–91, 2007.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. *Technical Report CRG-TR-93-1. Dept. of Computer Science, University of Toronto.*, 1993.

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 06 2012.

Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Giorgio Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3): 378 – 384, 1981.

Sebastian Reich. A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51(1):235–249, 2011.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144: 1–38, 2011.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

Gareth O. Roberts and Richard L. Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110, 1996a.

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

Peter J. Rossky, Jimmie D. Doll, and Harold L. Friedman. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

Havard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, 2005.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1—-96, 2018.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112, 2017.

Renata Sotirov. On solving the densest k-subgraph problem on large graphs. *Optimization Methods and Software*, 35(6):1160–1178, 2020.

Santosh S. Vempala. Recent progress and open problems in algorithmic convex geometry. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8, pages 42–64, 2010.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, page 681–688, 2011.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151:3–34, 2015.

Difan Zou and Quanquan Gu. On the convergence of Hamiltonian Monte Carlo with stochastic gradients. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 13012–13022, 2021.

Difan Zou, Pan Xu, and Quanquan Gu. Subsampled stochastic variance-reduced gradient Langevin dynamics. In *UAI 2018*, 2018a.

Difan Zou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced Hamilton Monte Carlo methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 6028–6037, 2018b.