Random Coordinate Underdamped Langevin Monte Carlo

Zhiyan Ding
Department of Mathematics
University of Wisconsin-Madison
zding49@math.wisc.edu

Qin Li
Department of Mathematics
University of Wisconsin-Madison
qinli@math.wisc.edu

Jianfeng Lu
Mathematics Department
Duke University
jianfeng@math.duke.edu

Stephen J. Wright
Department of Computer Sciences
University of Wisconsin-Madison
swright@cs.wisc.edu

Abstract

The Underdamped Langevin Monte Carlo (ULMC) is a popular Markov chain Monte Carlo sampling method. It requires the computation of the full gradient of the log-density at each iteration, an expensive operation if the dimension of the problem is high. We propose a sampling method called Random Coordinate ULMC (RC-ULMC), which selects a single coordinate at each iteration to be updated and leaves the other coordinates untouched. We investigate the computational complexity of RC-ULMC and compare it with the classical ULMC for strongly log-concave probability distributions. We show that RC-ULMC is always cheaper than the classical ULMC, with a significant cost reduction when the problem is highly skewed and high dimensional. Our complexity bound for RC-ULMC is also tight in terms of dimension dependence.

1 Introduction

Langevin Monte Carlo (LMC) is a popular Monte Carlo sampling method, widely used in Bayesian statistics and machine learning (Andrieu et al., 2003). The goal is to construct a Markov chain that approximately generates i.i.d. samples from a target distribution given by (with some abuse of notation, we do not distinguish a distribution with its density)

$$p_{\mathrm{X}}(x) = \frac{1}{Z} e^{-f(x)}, \tag{1}$$

where Z is a normalizing constant that ensures $\int p_{\mathrm{X}}(x)\,\mathrm{d}x = 1$. Throughout the paper we assume $f(x)$ is a convex function on $\mathbb{R}^d$, and thus $p_{\mathrm{X}}(x)$ is a log-concave probability distribution.

Among the many Monte Carlo sampling methods, LMC (Rossky et al., 1978; Parisi, 1981; Roberts and Tweedie, 1996) stands out for its simplicity: For each iteration one updates the location of the particle by descending along the gradient and adding properly scaled Gaussian noise. For strongly log-concave distributions, it has been established in recent years that the empirical distribution of the iterate in LMC converges exponentially fast to the target distribution, with total computational cost $\widetilde{O}(d^2/\epsilon^2)$ to achieve ϵ accuracy in Wasserstein distance (Dalalyan and Karagulyan, 2019; Durmus et al., 2019). Here and throughout the paper, we measure "cost" in terms of the total number of evaluations of a single element of the gradient, and assume that a full gradient evaluation requires about d times as much computation as a single component of the gradient.

To reduce the computational cost of sampling, the underdamped version of Langevin dynamics has recently been used to design the ULMC algorithm. By augmenting the state space with velocity variables, the underdamped Langevin dynamics converges to the equilibrium faster than the classical Langevin dynamics (Villani, 2006; Dolbeault et al., 2009; Baudoin, 2016; Monmarché, 2018; Cao et al., 2019). As a consequence, ULMC, the discrete version of the underdamped Langevin dynamics, also achieves faster convergence than LMC, the discrete version of the classi-

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

cal Langevin dynamics (Cheng et al., 2018a; Dalalyan and Riou-Durand, 2018; Eberle et al., 2018). To be more specific, to obtain ϵ-accuracy in sampling from a log-concave p_{X}, the computational complexity of ULMC is $\widetilde{O}(d^{3/2}/\epsilon)$, whereas that of LMC is $\widetilde{O}(d^2/\epsilon^2)$. This is an improvement in both d and ϵ.

This work aims at further improving the algorithm in terms of its dimension dependence, especially for the very high dimensional problems that arise often in practical applications (see (Ding et al., 2020b) for discussions of several examples). For these problems, ULMC requires a full evaluation of the gradient ∇f at each iteration, which often costs a factor of $O(d)$ greater than evaluating of a single component of the gradient. This factor arises when the expression of ∇f is not known explicitly, such as in partial differential equation (PDE) based inverse problems, where f is given implicitly by solving the forward problem given as a PDE, and finite-difference approximation to the full gradient would be d times more expensive than a single component. While automatic differentiation techniques have been developed, the cost of evaluation of the gradient often still leads to formidable computational and memory cost.

Other examples in which there is a factor-of-d difference in evaluation cost between a full gradient and single component of the gradient come from problems with particular structures, such as graph-based problems. Given a graph with nodes $\mathcal{N} = \{1, 2, \ldots, d\}$ and directed edges $\mathcal{E} \subset \{(i,j) : i,j \in \mathcal{N}\}$, 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 both partial derivatives of each component function f_{ij} for *all* edges in the graph, leading to a factor-of-d difference in evaluation cost.

In this work, we target these problems in which single components of the gradient are much less expensive than full gradients by incorporating the random coordinate descent (RCD) method from optimization into underdamped Langevin sampling algorithm. RCD differs from gradient descent (GD) in that it updates just a single component, chosen at random, of the variable vector x at each iteration. It takes a step in the negative gradient direction in just this component, leaving other components unchanged. (By contrast, gradient descent takes a step along the full negative gradient direction.) When there is a factor-of-d difference in cost between evaluating the full gradient and a single component of the gradient, worst-case bounds for convex problems are better for RCD than for GD, the cost reduction being particularly significant when the dimension d is high and f is "skewed" in a sense to be defined later. Specifically, it was shown in Nesterov (2012) that when the coordinate is chosen from a distribution weighted according to the directional Lipschitz constants, the complexity is reduced from $d\kappa$ to $d\kappa_{\max}$, where κ and $\kappa_{\max}$ are conditioning of f and the maximum directional conditioning of f respectively. Since $\kappa_{\max} \leq \kappa$ for all functions (Wright, 2015), RCD is always cheaper than GD. Further, when the dimension of the problem is high and f is skewed in the sense that $\kappa \approx d\kappa_{\max}$, the reduction in cost approaches a factor of d.

In this paper, we propose the random coordinate underdamped Langevin Monte Carlo (RC-ULMC) algorithm. We aim to improve the convergence of ULMC by utilizing cheaper steps, as in RCD, so we establish non-asymptotic convergence rates for RC-ULMC and compare with classical ULMC. Our main results are as follows:

1. The convergence rate of RC-ULMC depends on directional conditioning; see Theorem 6.2.
2. Comparing with ULMC, RC-ULMC is always cheaper than the classical ULMC, the change being $d^{3/2}\kappa^{3/2} \to (d^{3/2}+\kappa)\kappa^{1/2}$. This cost reduction is significant when f is skewed and the dimension is high; see the discussion following Corollary 6.3.
3. The complexity bound of the RC-ULMC we obtain is tight in both d and ϵ; see Proposition 6.4.

The remainder of the paper is as follows. We review literature in Section 2 and summarize basic notations and assumptions in Section 3. In Section 4 we review ULMC and its convergence properties. In Section 5 we present our new method RC-ULMC. Our main results are presented in Section 6, where we discuss the non-asymptotic convergence rate, the numerical cost, the cost saving compared to the classical ULMC, and the tightness of the result. Computational results are presented in Section 7. Technical derivations and proofs appear in Supplementary Materials.

2 Related works

The non-asymptotic analysis of LMC and ULMC sampling methods has been an active area (Cheng et al., 2018a; Dalalyan and Riou-Durand, 2018; Dalalyan and

Karagulyan, 2019; Durmus et al., 2019); and it has been established that ULMC gives a faster convergence under the same log-concavity and smoothness assumptions on the distribution (Cheng et al., 2018a; Dalalyan and Riou-Durand, 2018). When ULMC is modified with a better discretization scheme, e.g., the random midpoint method, the computational complexity can be even further reduced (Shen and Lee, 2019; He et al., 2020).

For ULMC, it was established in (Dalalyan and Riou-Durand, 2018) that it achieves ϵ error in Wasserstein metric within $\tilde{O}\left(d^{1/2}\kappa^{3/2}/\epsilon\right)$ iterations, where $\tilde{O}$ hides log factors. The total cost of ULMC is therefore $\tilde{O}\left(d^{3/2}\kappa^{3/2}/\epsilon\right)$. In comparison, the RC-ULMC method proposed in this paper is always cheaper and the saving can be significant for highly skewed distributions in high dimension.

The combination of RCD and LMC (based on overdamped Langevin dyanmics) has been recently explored in works (Shen et al., 2019; Ding et al., 2020b). This algorithm will be referred to as RC-OLMC (where "O" stands for overdamped). Compared with their result, the method in this paper converges faster both in terms of d and ϵ, similar to the saving obtained going from LMC to ULMC. This will be discussed further in Section 6.

Alternative sampling strategies have been developed without using the full gradient ∇f at each step. A standard approach is the Random Walk Metropolis algorithm, which combines a random walk proposal with Metropolis-Hastings acceptance-rejection step (Hastings, 1970), and thus only uses f at each iteration. However, they are less efficient in high dimensions compared with gradient based methods (Mattingly et al., 2012; Pillai et al., 2012). There have been recent interests in ensemble based sampling methods, in particular in the context of data assimilation, inspired by the ensemble Kalman filter (Evensen, 2006), such as (Garbuno-Inigo et al., 2020; Iglesias et al., 2013). Unfortunately, none of these methods can be completely "gradient-free" and at the same time consistent for non-Gaussian distributions (Ding and Li, 2019a,b). To achieve consistency, one can try to incorporate weights to particles, as is done in importance sampling (Geweke, 1989) or sequential Monte Carlo (Doucet et al., 2001), however such methods often face the difficulty of high variance (Ding et al., 2020a).

When the log-density $f(x)$ has the form of $f(x) = \sum_{i=1}^N f_i(x)$, one can randomly select a representative ∇f_r as a stochastic approximation to the full gradient, where r is uniformly chosen from $\{1, \cdots, N\}$. This leads to the stochastic gradient Langevin Monte Carlo method (Welling and Teh, 2011). Note that in general we can write the full gradient as $\nabla f = \sum_{i=1}^d \partial_i f \boldsymbol{e}_i$ (where $\boldsymbol{e}_i$ is the unit vector in i-th direction), and thus RCD and stochastic gradient, while used for different setups, share some similarity in reducing the cost of gradient evaluation.

Throughout the paper we do assume the log-concavity of p_{X} (the convexity of f). For optimization problems, there are abundant works relaxing this assumption (Gelfand and Mitter, 1991; Raginsky et al., 2017), but to our knowledge, no results on non-asymptotic convergence rate has been obtained for Bayesian sampling problems. One exception is (Cheng et al., 2018b) where the authors nevertheless require the convexity of f outside a ball of finite-size.

3 Notations and assumptions

Throughout the paper we assume convexity and gradient Lipschitz continuity of f.

Assumption 3.1. *The function f is second-order differentiable and μ-strongly convex for some $\mu > 0$ and the gradient ∇f is L-Lipschitz. Specifically, we have: for all $x, x' \in \mathbb{R}^d$*

$$f(x) - f(x') - \nabla f(x')^\top (x - x') \geq \frac{\mu}{2}|x - x'|^2 \quad (2)$$

and

$$|\nabla f(x) - \nabla f(x')| \leq L|x - x'|\,. \quad (3)$$

Since the full gradient is Lipschitz continuous, so is its directional derivative. We denote directional Lipschitz constants by L_i, $i = 1, 2, \ldots, d$, meaning that

$$|\partial_i f(x + t\boldsymbol{e}_i) - \partial_i f(x)| \leq L_i|t|\,, \quad (4)$$

for any $i = 1, 2, \ldots, d$, any $x \in \mathbb{R}^d$, and any $t \in \mathbb{R}$.

Denote $\nabla^2 f$ the Hessian, then the assumption implies that

$$\mu I_d \preceq \nabla^2 f(x) \preceq L I_d, \quad |\partial_{ii} f(x)| \leq L_i\,,$$

where $\partial_{ii} f(x)$ is the (i,i)-element of $\nabla^2 f(x)$. We also define condition numbers:

$$\kappa = L/\mu \geq 1,\ \kappa_i = L_i/\mu \geq 1,\ \kappa_{\max} = \max_i \kappa_i\,. \quad (5)$$

As shown in Wright (2015), we have

$$\kappa_i \leq \kappa_{\max} \leq \kappa \leq d\kappa_{\max}\,. \quad (6)$$

We note that both inequalities, $\kappa_{\max} \leq \kappa$ and $\kappa \leq d\kappa_{\max}$ are sharp. If $\nabla^2 f$ is a diagonal matrix, then $L_{\max} = L$, both being the largest eigenvalue of $\nabla^2 f$, so that $\kappa_{\max} = \kappa$. This is the case when all coordinates are independent of each other, for example $f = \sum_i \lambda_i x_i^2$. On the other hand, if f is highly

skewed, such as $f = (\sum_i x_i)^2$, so $\nabla^2 f = \mathsf{e} \cdot \mathsf{e}^\top$ (where $\mathsf{e} = [1, 1, \ldots, 1]^\top$), then $L = dL_{\max}$ and $\kappa = d\kappa_{\max}$.

Furthermore, since $\kappa_i \geq 1$, we have for $p > 0$ that

$$(d-1) + \kappa_{\max}^p \leq \sum_{i=1}^d \kappa_i^p \leq d\kappa_{\max}^p \leq d\kappa^p \,.$$

Both bounds are tight. In the case when $f = \kappa_1 |x_1|^2 + \sum_{i=1}^d |x_2|^2$ with $\kappa_1 > 1$, $\sum_{i=1}^d \kappa_i^p = (d-1) + \kappa_1^p = (d-1) + \kappa_{\max}^p$. On the other hand, when $f = \sum_i x_i^2$, we have $\kappa_i = \kappa_{\max} = 1$, then $\sum_{i=1}^d \kappa_i^p = d\kappa_{\max}^p = d\kappa^p$. And we say f is highly skewed if

$$\sum_{i=1}^d \kappa_i^p \approx (d-1) + \kappa_{\max}^p \,. \tag{7}$$

The extreme example in this setting is to set $f(x) = \frac{1}{2}\left(dx_1^2 + \sum_{i=2}^d x_i^2\right)$ with $d \gg 1$. In this example, $\kappa_{\max} = d$ and $\sum_{i=1}^d \kappa_i^p = (d-1) + \kappa_{\max}^p$.

To measure the distance between two probability distributions, we use the Wasserstein distance.

Definition 3.1. *The Wasserstein distance W_p (for any $p \geq 1$) between probability measures μ and ν is defined as*

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

where $\Gamma(\mu, \nu)$ is the set of distribution of $(X, Y) \in \mathbb{R}^{2d}$ whose marginal distributions, for X and Y respectively, are μ and ν.

In this paper, we will use the 2-Wasserstein metric W_2.

4 Classical ULMC

Underdamped Langevin dynamics is characterized by the following SDE:

$$\begin{cases} \mathrm{d}X_t = V_t \,\mathrm{d}t \,, \\ \mathrm{d}V_t = -2V_t \,\mathrm{d}t - \gamma \nabla f(X_t) \,\mathrm{d}t + \sqrt{4\gamma} \,\mathrm{d}B_t \,, \end{cases} \tag{8}$$

where $\gamma > 0$ is a parameter to be tuned, and B_t is the Brownian motion. Here we use the parametrization form of Cheng et al. (2018a) (alternative parametrizations are used in Dalalyan and Riou-Durand (2018); Shen and Lee (2019)). Denoting by $q(x, v, t)$ the probability density function of (X_t, V_t), we have that q satisfies the Fokker-Planck equation

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

It is well known that under mild conditions, q converges to $p(x, v) \propto \exp(-(f(x) + |v|^2/2\gamma))$ (see e.g., Villani (2006); Dolbeault et al. (2009); Baudoin (2016); Cao et al. (2019)), and thus the marginal density function for x becomes the target distribution $p_{\mathrm{X}}(x)$.

Denoting by $h > 0$ the time stepsize, we have that for $t \in [mh, (m+1)h]$, (8) is equivalent to

$$\begin{aligned} X(t) =& X(mh) + \frac{1 - e^{-2t}}{2} V(mh) \\ & - \frac{\gamma}{2} \int_{mh}^t \left(1 - e^{-2(t-s)}\right) \nabla f\left(X(s)\right) \,\mathrm{d}s \\ & + \sqrt{\gamma} \int_{mh}^t \left(1 - e^{-2(t-s)}\right) \,\mathrm{d}B_s \,, \end{aligned} \tag{9}$$

$$\begin{aligned} V(t) =& V(mh) e^{-2t} - \gamma \int_{mh}^t e^{-2(t-s)} \nabla f\left(X(s)\right) \,\mathrm{d}s \\ & + \sqrt{4\gamma} e^{-2t} \int_{mh}^t e^{2s} \,\mathrm{d}B_s \,. \end{aligned} \tag{10}$$

The sampling method, ULMC, can be viewed as a numerical solver for (9)-(10) based on the Euler approximation. Denoting by (x^m, v^m) the numerical approximation to $(X(mh), V(mh))$, and replacing $X(s)$ in (9)-(10) by x^m, Euler approximation yields that $(x^{m+1}, v^{m+1}) \in \mathbb{R}^{2d}$ are two Gaussian random vectors with the following expectation and covariance:

$$\begin{aligned} \mathbb{E}\, x^{m+1} &= x^m + \frac{1}{2}\left(1 - e^{-2h}\right) v^m \\ &\quad - \frac{\gamma}{2}\left(h - \frac{1}{2}\left(1 - e^{-2h}\right)\right) \nabla f(x^m) \,, \\ \mathbb{E}\, v^{m+1} &= v^m e^{-2h} - \frac{\gamma}{2}\left(1 - e^{-2h}\right) \nabla f(x^m) \,, \\ \mathrm{Cov}\left(x^{m+1}\right) &= \gamma \left[h - \frac{3}{4} - \frac{1}{4} e^{-4h} + e^{-2h}\right] \cdot I_d \,, \\ \mathrm{Cov}\left(v^{m+1}\right) &= \gamma \left[1 - e^{-4h}\right] \cdot I_d \,, \\ \mathrm{Cov}\left(x^{m+1}, v^{m+1}\right) &= \frac{\gamma}{2} \left[1 + e^{-4h} - 2e^{-2h}\right] \cdot I_d \,. \end{aligned} \tag{11}$$

Here $\mathbb{E}$ denotes the expectation, and $\mathrm{Cov}(a, b)$ denotes the covariance of a and b (abbreviated to $\mathrm{Cov}(a)$ when $b = a$), and I_d is the identity matrix in $\mathbb{R}^d$. We thus draw x^{m+1}, v^{m+1} from this Gaussian distribution numerically to update the iteration. We summarize ULMC in Algorithm 1.

The algorithm converges exponentially when f is strongly convex with Lipschitz continuous gradient; see Dalalyan and Riou-Durand (2018). The original statement uses a different parametrization. We translate the result to the current one in Supp. 1 and restate the result here.

Theorem 4.1. *[(Dalalyan and Riou-Durand, 2018, Theorem 2)] Assume f satisfies Assumption 3.1 and*

Algorithm 1 Underdamped Langevin Monte Carlo (ULMC)

Input: h (time stepsize); γ (parameter); d (dimension); $\nabla f(x)$; M (stopping index).
Initial: (x^0, v^0) i.i.d. sampled from the initial distribution $q_0(x, v)$.
for $m = 0, 1, \cdots, M$ **do**
1. Compute the expectation and the covariance as in (11).
2. Sample (x^{m+1}, v^{m+1}) from the associated Gaussian distribution.
end for
Output: $\{x^m\}$.

that $\gamma \le \frac{4}{\mu+L}$, $\quad h \le \frac{\gamma^{1/2}\mu}{8L}$. *Then we have*

$$W_m \le \sqrt{2}\exp(-0.375\mu h\gamma^{1/2}m)W_0 + (2d)^{1/2}\kappa h\,. \quad (12)$$

Here $W_m := W_2(q_m, p)$ *and* $q_m(x, v)$ *denotes the probability density function of iteration* m *of ULMC. Moreover, suppose the initial* W_0 *is* $O(1)$*, then the total number of iterations to achieve* ϵ *accuracy is* $\tilde{O}\left(\frac{d^{1/2}\kappa^{3/2}}{\mu^{1/2}\epsilon}\right)$*, and the cost is* $\tilde{O}\left(\frac{d^{3/2}\kappa^{3/2}}{\mu^{1/2}\epsilon}\right)$.

The cost depends on both the dimensionality d and condition number κ with 3/2 power for both.

5 Randomized Coordinate Underdamped Langevin Monte Carlo

We integrate the RCD idea into ULMC to yield our method RC-ULMC. Instead of updating every entry of the process as is done in (9)–(10), we randomly select one direction $r^m \in \{1, 2, \ldots, d\}$ and evolve only $(X_{r^m}, V_{r^m})(t)$. Correspondingly, we would only change one single entry $(x^m_{r^m}, v^m_{r^m})$ according to expectation and covariance, analogous to (11).

We denote the discrete distribution from which r^m is chosen by Φ, with ϕ_i being the probability of component i being chosen, that is,

$$\Phi := \{\phi_1, \phi_2, \ldots, \phi_d\}\,, \quad (13)$$

where $\phi_i > 0$ for all i and $\sum_{i=1}^d \phi_i = 1$. Denoting by h_i the stepsize when i-th direction is chosen, we choose h_i to be inversely proportional to ϕ_i, as follows:

$$h_i = \frac{h}{\phi_i}\,, \quad i = 1, 2, \ldots, d\,, \quad (14)$$

where $h > 0$ is a parameter that can be viewed as the expected stepsize. We also define the total elapsed time after m steps as

$$T^m = \sum_{n=0}^{m-1} h_{r^n}\,.$$

The initial iterate (x^0, v^0) is drawn from a distribution q_0, which can be any distribution that is easy to draw from (e.g., a normal distribution).

Because only component r^m is updated at iteration m, we have for $t \in [T^m, T^{m+1}]$ that

$$\begin{aligned} X_{r^m}(t) &= X_{r^m}(T^m) + \frac{1-e^{-2t}}{2}V_{r^m}(T^m) \\ &\quad - \frac{\gamma}{2}\int_{T^m}^t \left(1 - e^{-2(t-s)}\right)\partial_{r^m} f(X(s))\,\mathrm{d}s \\ &\quad + \sqrt{\gamma}\int_{T^m}^t \left(1 - e^{-2(t-s)}\right)\mathrm{d}B_s\,, \quad (15) \\ V_{r^m}(t) &= V_{r^m}(T^m)e^{-2(t-T^m)} \\ &\quad - \gamma\int_{T^m}^t e^{-2(t-s)}\partial_{r^m} f(X(s))\,\mathrm{d}s \\ &\quad + \sqrt{4\gamma}\int_{T^m}^t e^{-2(t-s)}\,\mathrm{d}B_s\,, \quad (16) \\ X_i(t) &= X_i(T^m),\ V_i(t) = V_i(T^m)\,, \quad i \ne r^m\,. \quad (17) \end{aligned}$$

We call (15)-(17) Random Coordinate Underdamped Langevin dynamics (RC-ULD). To obtain a practical algorithm, we apply the Euler approximation to these dynamics. Denoting by $(x^m_{r^m}, v^m_{r^m})$ the numerical approximation to $(X(T^m), V(T^m))$, we replace $\partial_{r^m} f(X(s))$ in (15)-(17) by $\partial_{r^m} f(x^m)$, so that $x_i^{m+1} = x_i^m$, $v_i^{m+1} = v_i^m$ for $i \ne r^m$, and $(x^{m+1}_{r^m}, v^{m+1}_{r^m})$ are two Gaussian random variables with the following expectation and covariance:

$$\begin{aligned} &\mathbb{E}x^{m+1}_{r^m} = x^m_{r^m} + \frac{1}{2}\left(1 - e^{-2h_{r^m}}\right)v^m_{r^m} \\ &\qquad - \frac{\gamma}{2}\left(h_{r^m} - \frac{1}{2}\left(1 - e^{-2h_{r^m}}\right)\right)\partial_{r^m} f(x^m)\,, \\ &\mathbb{E}v^{m+1}_{r^m} = v^m_{r^m}e^{-2h_{r^m}} - \frac{\gamma}{2}\left(1 - e^{-2h_{r^m}}\right)\partial_{r^m} f(x^m)\,, \\ &\mathrm{Cov}\left(x^{m+1}_{r^m}\right) = \gamma\left[h_{r^m} - \frac{3}{4} - \frac{1}{4}e^{-4h_{r^m}} + e^{-2h_{r^m}}\right]\,, \\ &\mathrm{Cov}\left(v^{m+1}_{r^m}\right) = \gamma\left[1 - e^{-4h_{r^m}}\right]\,, \\ &\mathrm{Cov}\left(x^{m+1}_{r^m}, v^{m+1}_{r^m}\right) = \frac{\gamma}{2}\left[1 + e^{-4h_{r^m}} - 2e^{-2h_{r^m}}\right]\,. \end{aligned} \quad (18)$$

Then, $(x^{m+1}_{r^m}, v^{m+1}_{r^m})$ is drawn according to this Gaussian distribution for the update. We summarize the RC-ULMC approach in Algorithm 2.

6 Main results

We have three main results regarding the underlying dynamics (RC-ULD), and the RC-ULMC algorithm.

Algorithm 2 Random Coordinate Underdamped Langevin Monte Carlo (RC-ULMC)

Input: h (time stepsize); γ (parameter); d (dimension); $\nabla f(x)$; probability set $\Phi := \{\phi_1, \phi_2, \ldots, \phi_d\}$; M (stopping index).
Initial: (x^0, v^0) i.i.d. sampled from the initial distribution induced by $q_0(x, v)$.
for $m = 0, 1, \cdots, M$ **do**
1. Draw r randomly from $1, 2, \ldots, d$ according to Φ;
2. Update (x^{m+1}, v^{m+1}) as follows:
- $x_i^{m+1} = x_i^m, v_i^{m+1} = v_i^m$ for $i \neq r$;
- sample (x_r^{m+1}, v_r^{m+1}) as Gaussian variables according to (18).

end for
Output: $\{x^m\}$.

In Section 6.1, we discuss convergence of the RC-ULD (15)-(17). This SDE can be viewed as the continuum version of the RC-ULMC algorithm. Only with the convergence of this SDE can we hope for the convergence of RC-ULMC. In Section 6.2, we describe the non-asymptotic convergence properties of RC-ULMC. From this result, we can determine an optimal strategy for selecting the coordinate r^m at each iteration. We will also compare our results with those for classical ULMC, showing that the bounds for RC-ULMC are always better. Moreover, when f is highly skewed — for example when $\kappa_1 = \kappa_{\max} \gg 1$ and $\kappa_i \approx 1$ for $i \geq 2$ — the total cost is $\widetilde{O}\left(\left(d^{3/2} + \kappa_{\max}\right)\kappa^{1/2}/\epsilon\right)$, as compared to $\widetilde{O}\left(d^{3/2}\kappa^{3/2}/\epsilon\right)$ for ULMC.

We provide an example in Section 6.3 to show that our bounds are tight.

6.1 Convergence of RC-ULD

Our first result is on the convergence of (15)-(17), the RC-ULD that incorporates random coordinate selection.

Denote $X^m = X(T^m)$, $V^m = V(T^m)$ and denote the probability filtration by $\mathcal{F}^m = \left\{x^0, v^0, r^{n \leq m}, B_{s \leq T^m}\right\}$. Then we have the following result about its geometric ergodicity.

Theorem 6.1. *Suppose that f satisfies Assumption 3.1 and $\gamma \leq \frac{1}{L}, h \leq \frac{\gamma\mu \min\{\phi_i\}}{312+12\gamma+8L+432L^2}$, then $\{(X^m, V^m)\}_{m=0}^{\infty}$ is a Markov chain. Denoting by $q_m(x, v)$ the probability density function of (X^m, V^m), we have the following:*

- *The stationary distribution has density $p(x, v) \propto \exp(-(f(x) + |v|^2/2\gamma))$.*
- *When the initial distribution q_0 has finite second moments, there exist constants $R > 0$ and $r > 1$ independent of m such that*

$$\int_{\mathbb{R}^{2d}} |q_m(x, v) - p(x, v)| \, \mathrm{d}x \, \mathrm{d}v \leq R r^{-m} . \quad (19)$$

The proof, which can be found in Supp. 3, uses the convergence analysis framework of (Mattingly et al., 2002), based on construction of a special Lyapunov function. This theorem suggests that the TV distance between q_m and p decays exponentially, meaning that (X^m, V^m) can be seen to be drawn from the target distribution p as $m \to \infty$. Since the RC-ULMC algorithm is its Euler approximation, the samples generated by this algorithm are drawn from p as well — approximately, up to a discretization error.

Note that R and r are independent of m in Theorem 6.1, but we do not have explicit control on its dependence on parameters such as h, d, and L. This is worse in comparison with the results in (Cheng et al., 2018a; Cao et al., 2019) for the underdamped Langevin dynamics, where the convergence rate is characterized explicitly in terms of all parameters. The difficulty of our case comes mainly from the complicated process of coordinate selection, which prevents us from applying the synchronous coupling approach of Cheng et al. (2018a); Dalalyan and Karagulyan (2019) directly to the dynamics (15)–(17) to establish contraction. Whether the hypocoercity estimate of Cao et al. (2019) can be applied remains an interesting future research direction.

6.2 Convergence of RC-ULMC

Regarding the non-asymptotic error analysis of RC-ULMC, we have the following result (cf. Theorem 4.1).

Theorem 6.2. *Suppose that f satisfies Assumption 3.1 and that*

$$\gamma \leq \frac{1}{L}, \quad h \leq \min\left\{\frac{\gamma\mu \min\{\phi_i\}}{240}\right\} . \quad (20)$$

Denote $q_m(x, v)$ the probability density function of iteration m of RC-ULMC and define $W_m = W_2(q_m, p)$. Then we have

$$W_m \leq 4 \exp\left(-\frac{\mu\gamma m h}{8}\right) W_0 + 40\gamma^{1/2} h \sqrt{\sum_{i=1}^{d} \frac{\kappa_i^2}{\phi_i^2}} . \quad (21)$$

The proof can be found in Supp. 4. This result indicates that the Wasserstein distance between q_m and the target distribution decays exponentially except for an error term of size $O(h)$. The convergence rate is given by $\mu\gamma$, and with the choice $\gamma = 1/L$, this quantity is the inverse condition number $1/\kappa$ of the objective function (see (5)). The second term in (21) reflects

the discretization error, with its size being determined by the directional condition number κ_i (see (5)) and the random selection probability distribution Φ.

This theorem not only allows us to estimate the number of iterations required to achieve a preset accuracy, but also suggests that we choose $\{\phi_i\}$ in a way that minimizes the bound.

Corollary 6.3. *Suppose that the conditions of Theorem 6.2 hold and $\gamma = 1/L$. We have the following estimates.*

- *For any $\epsilon > 0$, the number of needed iterations M to attain $W_M \leq \epsilon$ is*

$$M = \Theta\left(\frac{\kappa^{1/2}\sqrt{\sum_{i=1}^d \kappa_i^2/\phi_i^2}}{\mu^{1/2}\epsilon}\log\left(\frac{W_0}{\epsilon}\right)\right). \quad (22)$$

- *The optimal choice of ϕ_i is*

$$\phi_i = \frac{L_i^{2/3}}{\sum_{j=1}^d L_j^{2/3}}, \quad i = 1, 2, \ldots, d. \quad (23)$$

In this case, the number of iterations required is

$$M = \Theta\left(\frac{\kappa^{1/2}\left(\sum_{j=1}^d \kappa_j^{2/3}\right)^{3/2}}{\mu^{1/2}\epsilon}\log\left(\frac{W_0}{\epsilon}\right)\right). \quad (24)$$

The proof can be found in Supp. 5. Suppose the objective function f is well-conditioned in every direction, so that $\kappa_i = O(1)$ for all i. Then according to both (22) and (24), we see that the cost is roughly $\widetilde{O}(d^{3/2}/\epsilon)$. This order is the same as for the classical ULMC shown in Theorem 4.1.

When f is not as well-conditioned, meaning κ_i are not uniformly small in every direction, then RC-ULMC can have a significant advantage over the classical ULMC. In practice, if we have some a priori estimate of κ_i, we can choose the optimal Φ and the cost estimate will be given by (24). Of course, such a priori information might not be available, in such case, we can choose uniformly the coordinate at each iteration: $\phi_i = 1/d$. We compare below the cost of RC-ULMC in these two scenarios with $\widetilde{O}(d^{3/2}\kappa^{3/2}/\mu^{1/2}\epsilon)$, the cost of the classical ULMC, as shown in Theorem 4.1.

Case 1: Uniform sampling, where we choose $\phi_i = 1/d$. From (22), we found that the cost is

$$\widetilde{O}\left(\frac{d\kappa^{1/2}\sqrt{\sum_{i=1}^d \kappa_i^2}}{\mu^{1/2}\epsilon}\right), \quad (25)$$

where the $\widetilde{O}$ ignores log terms. Since $\sum_{i=1}^d \kappa_i^2 \leq d\kappa_{\max}^2 \leq d\kappa^2$, we observe that RC-ULMC is always cheaper than ULMC. Furthermore, when f is highly skewed (7) with $p = 2$, then (25) is reduced to $\tilde{O}\left(\frac{d(\sqrt{d}+\kappa_{\max})\kappa^{1/2}}{\mu^{1/2}\epsilon}\right)$. This bound indicates that RC-ULMC is significantly cheaper than ULMC when both d and κ are large.

Case 2: With the optimal choice of Φ (using (23)), the cost of RC-ULMC is equivalent to M in (24). We still have that RC-ULMC is always cheaper than ULMC. Furthermore, when f is highly skewed (7) with $p = 3/2$, then we have upper bound $\left(\sum_{j=1}^d \kappa_j^{2/3}\right)^{3/2} \approx \left((d-1) + \kappa_{\max}^{2/3}\right)^{3/2} \leq 2(d^{3/2} + \kappa_{\max})$. By substituting this bound into (24), we get the following bound:

$$\tilde{O}\left(\frac{(d^{3/2} + \kappa_{\max})\kappa^{1/2}}{\mu^{1/2}\epsilon}\right). \quad (26)$$

The reduction over ULMC is significant when either d or κ is large.

We also compare RC-ULMC with RC-OLMC discussed in (Shen et al., 2019; Ding et al., 2020b). To achieve ϵ-accuracy, the total cost of RC-OLMC is $\widetilde{O}\left(\frac{\sum_{i=1}^d \kappa_i^2/\phi_i}{\mu\epsilon^2}\right)$. Noting that $\sqrt{\sum_{i=1}^d \kappa_i^2/\phi_i^2} \leq \sum_{i=1}^d \kappa_i/\phi_i \leq \sum_{i=1}^d \kappa_i^2/\phi_i$, (22) is always smaller, meaning RC-ULMC is always cheaper than RC-OLMC when $\epsilon < 1/L^{1/2}$. Furthermore, if we choose uniform sampling ($\phi_i = 1/d$) and assume γ, μ, κ_i are $O(1)$, then the cost of RC-ULMC is $\widetilde{O}\left(d^{3/2}/\epsilon\right)$ while the cost of RC-OLMC is $\widetilde{O}\left(d^2/\epsilon^2\right)$. We have a significant improvement in both d and ϵ.

Finally, we note that in (Shen and Lee, 2019), randomzied midpoint method (RMM) is used to discretize SDE (8). According to (He et al., 2020), RMM needs $\widetilde{O}(d^{1/3}\kappa/\epsilon^{2/3})$ iteration steps to achieve ϵ-accuracy, which equates to a cost of $\widetilde{O}(d^{4/3}\kappa/\epsilon^{2/3})$. By comparison, (26) is smaller in some extreme regimes, such as $d^{2/9}\epsilon^{-2/3} < \kappa < \epsilon^{2/3}d^{8/3}$. We note that the comparison between RC-ULMC and RMM is not entirely fair since RMM uses a better discretization scheme than the Euler approximation (11) used in RC-ULMC. It is possible to include the RCD idea to RMM on ULMC as well, for a potentially better convergence rate. We leave that topic to future investigation.

6.3 Tightness of the bound

Corollary 6.3 shows that the numerical cost is roughly $\widetilde{O}(d^{3/2}/\epsilon)$ when the problem is well conditioned. We show by use of an example that this bound is tight with respect to d and ϵ.

Proposition 6.4. *Let the target distribution be a standard Gaussian* $p_{\mathrm{X}}(x) = \frac{1}{(2\pi)^{d/2}}\exp(-|x|^2/2)$, *which is the marginal distribution of the target distribution* $p(x,v) = \frac{1}{(2\pi)^d}\exp(-|x|^2/2 - |v|^2/2)$. *Suppose the initial distribution* q_0 *is chosen to be* $q_0(x,v) = \frac{1}{(2\pi)^d}\exp(-|x-u|^2/2 - |v|^2/2)$ *with* $u \in \mathbb{R}^d$ *and* $u_i = 1/400$ *for all* i. *Then if we choose* h *so that* $h < 10^{-8}/d$, *we have*

$$W_m \geq \frac{\exp\left(-4hm\right)}{800^2}\frac{d}{8} + \frac{d^{3/2}h}{320 - 464dh}, \tag{27}$$

where $W_m := W_2(q_m, p)$, *and* q_m *is the probability distribution of* (x^m, v^m) *generated by Algorithm 2 with* $\gamma = 1/L = 1$ *using uniform coordinate sampling. Furthermore, to have* $W_m \leq \epsilon$, *one needs at least* $M = \widetilde{O}(d^{3/2}/\epsilon)$ *iterations.*

The proof can be found in Supp. 6. For this particular initialization, we have $W_0 = \sqrt{d}/400$. According to (27), we can guarantee $W_m \leq \epsilon$ only if both terms are smaller than ϵ, meaning that (ignoring a log term) $h \lesssim \frac{320\epsilon}{d^{3/2}}, m \gtrsim \frac{1}{4h}$, which implies a cost of $\widetilde{O}(d^{3/2}/\epsilon)$.

7 Numerical experiments

We present numerical evidence to demonstrate the improvement of RC-ULMC over the classical ULMC. In this section we only present a toy problem, and we leave a more practical example to Supp. 2.

In the examples, we repeat the Markov chain for N independent trials and denote $\{x^{(i),M}\}_{i=1}^N$ the list of N samples at M-th iteration. Since Wasserstein distance is difficult to measure directly numerically, especially when the underlying distribution function is presented by a list of particles, we evaluate the following error as a surrogate:

$$\text{Error} = \left\| \frac{1}{N}\sum_{i=1}^{N}\psi(x^{(i),M}) - \mathbb{E}_{p_{\mathrm{X}}}(\psi) \right\|_2, \tag{28}$$

where $\psi(x)$ is a matrix-valued function and referred to as the test function, $\|\cdot\|_2$ means the spectral norm of the matrix, and $\mathbb{E}_{p_{\mathrm{X}}}(\psi)$ is the expected value of ψ with respect to the target distribution p_{X}.

In the first example, we set the target distribution function to be $p_{\mathrm{X}}(x) \propto p_1(x)p_2(x)$ with $p_2(x) = \exp\left(-\sum_{i=11}^{d}\frac{|x_i|^2}{2}\right)$ and $p_1(x) = \exp\left(-\frac{1}{2}\mathsf{x}^\top\Gamma^\top\Gamma\mathsf{x}\right)$, where $\mathsf{x} = (x_1, x_2, \ldots, x_{10})^\top$ is the list of first 10 entries, and $\Gamma = \mathsf{T} + \frac{d}{10}I$. Here I is the 10×10 identity matrix and T is a random matrix whose entries are i.i.d. standard Gaussian random variables. This example has an ill-conditioned f. The Lipschitz constants are $\mathbb{E}(L_{1\leq i\leq 10}) = \frac{d^2}{100}$ and $L_{i\geq 10} = 1$. Thus $\mu = 1$, and $\mathbb{E}(\kappa_{1\leq i\leq 10}) = \frac{d^2}{100}$. When $d \gg 1$, we have $\sum_{j=1}^{d}\kappa_i^p \ll d\kappa^p$ for $p \geq 1$.

In the simulation we set $d = 100$, $N = 10^5$, and let $\psi(x) = \mathsf{x}\mathsf{x}^\top \in \mathbb{R}^{10\times 10}$.

Initially, all particles are drawn from the density distribution $p_0(x,v) \propto p_1(\mathsf{x} - 0.5\mathsf{e}_{10})p_2(x)\exp\left(-|v|^2/2\right)$, where e_{10} is a vector in $\mathbb{R}^{10}$ and all entries equal to 1. It is expected that the density of the target distribution is $p(x,v) \propto p_{\mathrm{X}}(x)\exp\left(-|v|^2/2\right)$, making p_{X} the marginal probability density.

The result is plotted in Figure 1. To run RC-ULMC, we use time stepsize $h = 10^{-4}$. For comparison we also run ULMC, however, due to the cost difference per iteration, there is no standard choice of h for ULMC for a fair comparison. Since $d = 100$ in this example, per iteration, the cost of ULMC is 100 times of that of RC-ULMC, we first experiment ULMC with $h = 10^{-2}$. It is clear that RC-ULMC, presented by the purple line achieves a lower error than ULMC with the same amount of cost.

We then test ULMC with different choices of h, hoping to find its best performance. As one decreases h, the error plateau is also lower, but the decay rate of error with respect to the cost decreases, as one can see by comparing the yellow, red and blue lines in Figure 1, all produced by ULMC with different values of h. None of them, however, can compete with RC-ULMC regarding the level of error at the same cost.

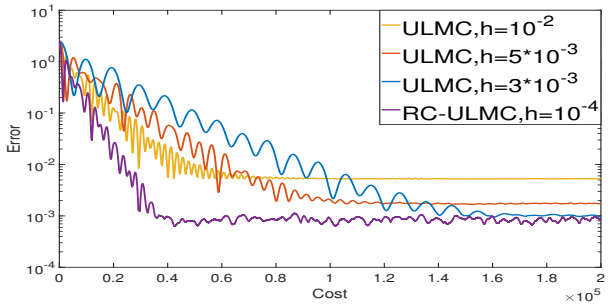

Figure 1: The decay of error with respect to the cost (the number ∂f calculations).

Acknowledgements

The work of JL is supported in part by National Science Foundation via grant NSF DMS-2012286 and CCF-1934964. The work of QL and ZD is supported in part by National Science Foundation via grant NSF-DMS 1750488 and Wisconsin Alumni Research Foundation. The work of QL, ZD and SW is supported NSF-TRIPODS 1740707 and NSF-TRIPODS 2023239.

References

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

Baudoin, F. (2016). Wasserstein contraction properties for hypoelliptic diffusions. *arXiv: 1602.04177*.

Cao, Y., Lu, J., and Wang, L. (2019). On explicit L^2-convergence rate estimate for underdamped Langevin dynamics. *arXiv:1908.04746*.

Cheng, X., Chatterji, N., Bartlett, P., and Jordan, M. (2018a). Underdamped Langevin MCMC: a non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 300–323.

Cheng, X., Chatterji, N. S., Abbasi-Yadkori, Y., Bartlett, P., and Jordan, M. I. (2018b). Sharp convergence rates for langevin dynamics in the nonconvex setting. *ArXiv*, abs/1805.01648.

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

Dalalyan, A. S. and Riou-Durand, L. (2018). On sampling from a log-concave density using kinetic Langevin diffusions. *arXiv:1807.09382*.

Ding, Z. and Li, Q. (2019a). Ensemble Kalman inversion: mean-field limit and convergence analysis. *arXiv:1908.05575*.

Ding, Z. and Li, Q. (2019b). Ensemble Kalman sampler: mean-field limit and convergence analysis. *arXiv:1910.12923*.

Ding, Z., Li, Q., and Lu, J. (2020a). Ensemble Kalman inversion for nonlinear problems: weights, consistency, and variance bounds. *Found. Data Sci.* arXiv:2003.02316.

Ding, Z., Li, Q., Lu, J., and Wright, S. J. (2020b). Random coordinate langevin monte carlo. *arXiv*, abs/2010.01405.

Dolbeault, J., Mouhot, C., and Schmeiser, C. (2009). Hypocoercivity for kinetic equations with linear relaxation terms. *Comptes Rendus Mathematique*, 347(9):511 – 516.

Doucet, A., Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer New York ; London.

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

Eberle, A., Guillin, A., and Zimmer, R. (2018). Couplings and quantitative contraction rates for langevin dynamics. *arXiv*, abs/1703.01617.

Evensen, G. (2006). *Data Assimilation: The ensemble Kalman filter*. Springer-Verlag.

Garbuno-Inigo, A., Hoffmann, F., Li, W., and Stuart, A. (2020). Interacting Langevin diffusions: Gradient structure and Ensemble Kalman sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1):412–441.

Gelfand, S. B. and Mitter, S. K. (1991). Recursive stochastic algorithms for global optimization in R^d. *SIAM Journal on Control and Optimization*, 29(5):999–1018.

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

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

He, Y., Erdogdu, M., and Balasubramanian, K. (2020). On the ergodicity, bias and asymptotic normality of randomized midpoint sampling method. *To appear in NeurIPS 2020*.

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

Mattingly, J., Stuart, A., and Higham, D. (2002). Ergodicity for sdes and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic Processes and their Applications*, 101(2):185 – 232.

Mattingly, J. C., Pillai, N. S., Stuart, A. M., et al. (2012). Diffusion limits of the random walk metropolis algorithm in high dimensions. *The Annals of Applied Probability*, 22(3):881–930.

Monmarché, P. (2018). Hypocoercivity in metastable settings and kinetic simulated annealing. *Probab. Theory Relat. Fields*, 172:1215–1248.

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

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

Pillai, N. S., Stuart, A. M., Thiéry, A. H., et al. (2012). Optimal scaling and diffusion limits for the langevin algorithm in high dimensions. *The Annals of Applied Probability*, 22(6):2320–2356.

Raginsky, M., Rakhlin, A., and Telgarsky, M. (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *COLT*.

Roberts, G. and Tweedie, R. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

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

Shen, L., Balasubramanian, K., and Ghadimi, S. (2019). Non-asymptotic results for langevin monte carlo: Coordinate-wise and black-box sampling. *arXiv*, abs/1902.01373.

Shen, R. and Lee, Y. T. (2019). The randomized midpoint method for log-concave sampling. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pages 2098–2109.

Villani, C. (2006). Hypocoercivity. *Memoirs of the AMS - American Mathematical Society*, 202.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688.

Wright, S. (2015). Coordinate descent algorithms. *Mathematical Programming*, 151:3–34.