Randomized Midpoint Method for Log-Concave Sampling

Dae Hyeun Cheong [*] Hangyu Li [†] Junwon Choi [‡] Shang Chen [§]

June 10, 2025

1 Introduction

In many statistical and machine learning problems, a central task is to compute the expected value of a function $\varphi(x)$ with respect to a probability distribution $\pi(x)$, given by

$$\mathbb{E}_\pi[\varphi(x)] = \int \varphi(x)\pi(x)\,dx.$$

However, in high-dimensional settings, the distribution $\pi(x)$ is often not known in normalized form[1]. Instead, it is typically specified up to a proportionality constant: $\pi(x) = \frac{1}{Z_f}e^{-f(x)}$, where $f(x)$ is a scalar-valued function (commonly referred to as a potential or energy function), and $Z_f = \int e^{-f(x)}dx$ is the normalizing constant. In high dimensions, evaluating Z_f is generally intractable due to the cost of integration over the entire domain, rendering direct computation of $\pi(x)$ infeasible.

To address this, modern sampling algorithms avoid computing Z_f explicitly by relying only on *ratios* of probabilities, in which the normalizing constant cancels out. As a result, it is common practice to operate directly on $f(x)$ and treat $\pi(x) \propto e^{-f(x)}$ as a target distribution. Recent work on sampling algorithms has focused on developing methods with provable guarantees for how efficiently they can approximate such target distributions.

A widely adopted assumption in this line of work is that $f(x)$ is *m-strongly convex* with an *L-Lipschitz continuous gradient*, so that $\pi(x)$ defines a *log-concave distribution*. This assumption is both theoretically appealing and practically useful: log-concave distributions include many commonly encountered examples in Bayesian posterior inference, statistical physics, and convex optimization. Moreover, the geometric structure imposed by convexity enables strong convergence guarantees for gradient-based sampling algorithms.

One of the most important reasons for ensuring that $\pi(x) \propto e^{-f(x)}$ is log-concave is that it enables the construction of continuous-time stochastic processes—specifically, stochastic differential equations (SDEs)—that use $\pi(x)$ as their stationary distribution. A stationary distribution is one that remains invariant under the dynamics of the process: if the process starts from this distribution, it remains in it for all future times. More importantly, for appropriately constructed SDEs and under standard regularity assumptions (such as smoothness and strong convexity of $f(x)$), the process is *ergodic*—that is, the distribution of its state converges to $\pi(x)$ regardless of the initial condition. This provides a principled way to generate approximate samples from $\pi(x)$: by

[*] UC Davis, dcheong@ucdavis.edu

[†] UC Davis, alnli@ucdavis.edu

[‡] UC Davis, jnwchoi@ucdavis.edu

[§] UC Davis, srochen@ucdavis.edu

[1] A probability density function (pdf) $\pi(x)$ is said to be *normalized* if it integrates to 1 over its domain, i.e., $\int \pi(x)\,dx = 1$.

simulating the trajectory of an SDE whose stationary distribution is $\pi(x)$, one can obtain samples that asymptotically follow the desired target. The log-concavity of $\pi(x)$ plays a critical role in this approach, as it ensures the geometric and analytic conditions needed for convergence and stability.

However, just applying an SDE directly is not possible in practice, because SDEs describe how a random variable evolves in continuous time—that is, how it changes at every infinitesimally small moment. Since we cannot simulate a truly continuous process on a digital computer, we need to approximate the SDE by a discrete-time process. This means we break the time interval into small steps and simulate the system only at those steps. This procedure is called *discretization*.

To discretize an SDE, we need to approximate its solution over short time intervals using simpler expressions. Traditionally, there are two common approaches. One is the *Taylor expansion method*, which estimates how the system evolves over a small time step by expanding the solution as a polynomial using derivatives at the starting point.

The other is the *collocation method*: instead of solving the SDE as a differential equation $x'(t) = F(x(t))$, we rewrite it as an equivalent integral equation. For instance, we can equivalently write the solution of SDE differential equation using Fundamental Theorem of Calculus as

$$x(t) = x(0) + \int_0^t F(x(s))\,ds \text{ for all } t \geq 0$$

This motivates defining an operator $\mathcal{T}$ that acts on functions x on at a chosen time point t as

$$\mathcal{T}(x)(t) = x(0) + \int_0^t F(x(s))\,ds.$$

The true solution $x(t)$ is a *fixed point* of this operator, meaning $x = \mathcal{T}(x)$. In practice, we approximate the solution by choosing an initial guess $x_0(t)$ and applying $\mathcal{T}$ repeatedly:

$$x_1 = \mathcal{T}(x_0), \quad x_2 = \mathcal{T}(x_1), \quad \ldots,$$

until the process stabilizes. In other words, if Taylor methods are like drawing small local steps using derivatives, collocation is like shaping the entire curve by repeatedly integrating its shape and re-evaluating it.

In the context of discretization, the goal is not to solve the SDE exactly, but to construct a discrete-time update rule that accurately approximates the behavior of the continuous process over short time intervals. The fixed-point formulation $\mathcal{T}(x) = x$ becomes useful here because it allows us to design numerical methods that approximate the solution path of the SDE by iteratively evaluating $\mathcal{T}$ or its approximation. Importantly, the accuracy of this discretization step is what ultimately determines how well the sampler can replicate the long-term behavior of the true continuous process.

Once we have a sufficiently accurate approximation to the path $x(t)$ over each small time interval, we can simulate the entire trajectory incrementally. Since the underlying SDE is designed to be ergodic, the distribution of the discretized process converges to the target stationary distribution $\pi(x) \propto e^{-f(x)}$ as the simulation progresses over time. In this context, the role of collocation in discretization is to ensure that each local update step closely follows the true continuous dynamics, thereby enabling asymptotically correct sampling behavior.

The *randomized midpoint method (RMM)* can be seen as a refined version of the collocation framework. The proposed method achieves the best theoretical guarantees using only two applications of the collocation operator $\mathcal{T}$ per step—equivalently, two gradient evaluations.

$$x(t_{n+\frac{1}{2}}) = x(t_n) + \int_{t_n}^{t_{n+\frac{1}{2}}} \nabla f(x(t_n))\,ds \text{ (First iteration: Finding midpoint)}$$

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} \nabla f(x(t_{n+\frac{1}{2}}))\, ds \text{ (Second iteration: Finding next sample collocation point)}$$

When applied to the underdamped Langevin diffusion (ULD), the RMM achieves a convergence rate of $\widetilde{O}\left(\frac{\kappa^{7/6}}{\epsilon^{1/3}} + \frac{\kappa}{\epsilon^{2/3}}\right)$, which improves upon all previously known algorithms by eliminating dependence on the dimension d, and reducing dependence on the condition number $\kappa = \frac{L}{m}$ and the accuracy parameter $\frac{1}{\epsilon}$. See Appendix A.1 for a detailed comparison with prior work.

1.1 Related Work: OLD and ULD

A wide range of sampling algorithms have been proposed for solving the log-concave sampling problem. Most follow a two-stage paradigm: (1) construct a continuous-time Markov process whose stationary distribution is $\pi(x) \propto e^{-f(x)}$; and (2) discretize this process in a way that preserves the convergence guarantees of the original dynamics.

Overdamped Langevin Diffusion (OLD). A canonical example of such a process is the overdamped Langevin diffusion (OLD), which is governed by the SDE

$$dx(t) = -\nabla f(x(t))\, dt + \sqrt{2}\, dB_t,$$

where B_t denotes standard Brownian motion[2]. This diffusion is ergodic and has $\pi(x)$ as its unique stationary distribution under standard assumptions such as strong convexity and smoothness of $f(x)$. When discretized via the Euler–Maruyama scheme[3], it leads to the Unadjusted Langevin Algorithm (ULA), which has been extensively studied. ULA can achieve ε-accuracy in $\widetilde{\mathcal{O}}(\kappa^2/\varepsilon^2)$ steps, where $\kappa = L/m$ is the condition number of f [3].

Underdamped Langevin Diffusion (ULD). The ULD is another popular method that improves upon OLD by incorporating second-order information. ULD augments the system with a momentum variable $v(t)$, evolving according to

$$\begin{aligned} dv(t) &= -2v(t)\, dt - u\nabla f(x(t))\, dt + 2\sqrt{u}\, dB_t, \\ dx(t) &= v(t)\, dt, \end{aligned}$$

where $u = 1/L$. This second-order structure allows ULD to retain inertia and explore the sample space more effectively, especially in ill-conditioned regions of the distribution. Unlike OLD, which updates position using local gradients and is prone to slow mixing, ULD introduces a momentum-based update that can cross high-curvature regions with greater ease. Even a basic Euler discretization of ULD can reach ε-accuracy in $\widetilde{\mathcal{O}}(\kappa^2/\varepsilon)$ steps [5].

[2] Brownian motion B_t is a continuous-time stochastic process with independent and normally distributed increments. It models random movement and introduces noise into the dynamics, allowing the process to explore the sample space.

[3] The Euler–Maruyama method is a first-order numerical integrator for stochastic differential equations (SDEs). It generalizes the classical Euler method for ODEs by accounting for both drift and diffusion terms. For an SDE of the form $dx(t) = a(x(t))\, dt + b(x(t))\, dB_t$, the method approximates the solution over a small time step h as

$$x_{k+1} = x_k + a(x_k)h + b(x_k)\sqrt{h} \cdot \xi_k,$$

where $\xi_k \sim \mathcal{N}(0, 1)$. It is widely used due to its simplicity and ease of implementation, although it introduces discretization bias.

For this reason, the paper focuses on discretizing ULD. Compared to OLD, ULD provides a compelling tradeoff between theoretical guarantees and implementation simplicity. Moreover, the continuous-time dynamics of ULD are governed by a contraction property in Wasserstein distance[4], which facilitate sharper convergence analysis[5]. [1].

Discretization: From Euler to Midpoint. A central challenge in sampling from continuous-time Langevin dynamics lies in discretizing the underlying SDE accurately. Traditional discretization methods often accumulate significant error over time because they rely on local, low-order approximations of the true solution trajectory. In particular, these methods evaluate the drift term (e.g., ∇f) only at the start or midpoint of each time interval, without accounting for how the state evolves within the interval. As a result, the discretized path can diverge from the continuous-time dynamics, leading to bias in the sampling distribution.

To overcome this, the randomized midpoint method introduces a novel collocation-based strategy that treats the solution to the SDE as a fixed point of an integral operator. Rather than explicitly converting the SDE into an ODE—which is often required in second-order methods like Hamiltonian Monte Carlo or some deterministic integrators—the method works directly with the integral formulation. This avoids the need to track the full velocity dynamics or solve auxiliary differential systems that can introduce additional numerical instability and computational overhead.

At each step, the randomized midpoint method selects a random interpolation point within the time interval and evaluates the drift at that point to approximate the integral of the vector field. This yields an unbiased estimator of the true drift, while maintaining low variance and requiring only one gradient evaluation per step. The randomized midpoint method outperforms both Euler-type and higher-order deterministic integrators in terms of both theoretical guarantees and empirical efficiency [7].

2 Methodology

The Randomized Midpoint Method (RMM) is a discretization scheme developed to accurately simulate time-continuous SDEs. While the method can be applied to a broader class of SDEs, this paper focuses specifically on its use in the context of ULD. At each iteration, RMM samples a random value $\alpha \in [0,1]$ to select a random interpolation point within the time interval, computes an intermediate position $x_{n+1/2}$, and updates both the position and velocity based on the drift evaluated at this randomly chosen point.

RMM is considered a superior discretization method compared to Euler-type methods, which inherently assume that the gradient evaluated at the beginning of the interval remains constant throughout. This assumption introduces systematic bias, especially when the drift function $\nabla f(x)$ changes significantly over the interval. In contrast, RMM mitigates this bias by evaluating the drift at a random location within the interval. This randomized evaluation yields an unbiased approximation of the integrated drift, leading to more accurate simulation of the underlying stochastic

[4] The Wasserstein distance is a way to measure how different two probability distributions are, taking into account the geometry of the underlying space. Intuitively, it represents the minimal čostöf transporting mass to transform one distribution into another, where cost is defined by how far the mass has to move. For two probability measures μ and ν on $\mathbb{R}^d$, the 2-Wasserstein distance is defined as:

$$W_2(\mu,\nu)^2 = \inf_{\gamma\in\Gamma(\mu,\nu)} \int \|x-y\|^2 \, d\gamma(x,y),$$

where $\Gamma(\mu,\nu)$ is the set of all couplings (joint distributions) with marginals μ and ν. See Appendix A.2

[5] In the context of sampling, contraction in Wasserstein distance means that as the stochastic process evolves, the distribution of the iterates gets closer to the target distribution in this geometric sense. This property is essential for proving strong convergence guarantees.

dynamics. (See Appendix A.6 for a formal justification of the unbiasedness of the Randomized Midpoint Method.)

Algorithm 1 Randomized Midpoint Method for ULD

```
procedure RandomMidpoint($x_0, v_0, N, h$)
    for $n = 0, \ldots, N-1$ do
        Randomly sample $\alpha$ uniformly from $[0, 1]$
        Generate Gaussian random variable $\left(W_1^{(n)}, W_2^{(n)}, W_3^{(n)}\right) \in \mathbb{R}^{3d}$ as in Appendix A.5
        $x_{n+\frac{1}{2}} = x_n + \frac{1}{2}\left(1 - e^{-2\alpha h}\right) v_n - \frac{1}{2} u \left(\alpha h - \frac{1}{2}(1 - e^{-2\alpha h})\right) \nabla f(x_n) + \sqrt{u} W_1^{(n)}$
        $x_{n+1} = x_n + \frac{1}{2}\left(1 - e^{-2h}\right) v_n - \frac{1}{2} u h \left(1 - e^{-2(h-\alpha h)}\right) \nabla f(x_{n+\frac{1}{2}}) + \sqrt{u} W_2^{(n)}$
        $v_{n+1} = v_n e^{-2h} - u h e^{-2(h-\alpha h)} \nabla f(x_{n+\frac{1}{2}}) + 2\sqrt{u} W_3^{(n)}$
    end for
end procedure
```

Justification of Methodology The idea of RMM is built from ULD equations. If we take integral on $[0, h]$, we can get the exact solutions [6] for initial point (x_n,v_n):

$$x_n^*(h) = x_n + \frac{1 - e^{-2h}}{2} v_n - \frac{u}{2} \int_0^h \left(1 - e^{-2(h-s)}\right) \nabla f(x_n^*(s))\, \mathrm{d}s + \sqrt{u} \int_0^h \left(1 - e^{-2(h-s)}\right) \mathrm{d}B_s,$$

$$v_n^*(h) = v_n e^{-2h} - u \left(\int_0^h e^{-2(h-s)} \nabla f(x_n^*(s))\, \mathrm{d}s \right) + 2\sqrt{u} \int_0^h e^{-2(h-s)}\, \mathrm{d}B_s.$$

The problem is that $x_n^*(s)$ is unknown[7]. Therefore, we are also not able to get $\nabla f(x_n^*(s))$. The previous method dealing with this is to replace $\nabla f(x_n^*(s))$ with $\nabla f(x_n)$, the gradient of the initial point as an approximation. However, this method gives a relatively large error.

Here RMM identify an accurate estimator by sampling a random number α uniformly form [0,1] and then have αh as a random point in [0,h]. By doing so, we have $\int_0^h \left(1 - e^{-2(h-s)}\right) \nabla f(x_n^*(s)) \approx h\left(1 - e^{-2(h-\alpha h)}\right) \nabla f\left(x_n^*(\alpha h)\right)$. And the estimator $h\left(1 - e^{-2(h-\alpha h)}\right) \nabla f\left(x_n^*(\alpha h)\right)$ is unbiased [8]. Now, we bring the unbiased estimator back to the previous equation:

$$x_{n+1} = x_n + \frac{1 - e^{-2h}}{2} v_n - \frac{u}{2} h \left(1 - e^{-2(h-\alpha h)}\right) \nabla f\left(x_{n+\frac{1}{2}}\right) + \sqrt{u} \int_0^h \left(1 - e^{-2(h-s)}\right) \mathrm{d}B_s,$$

$$v_{n+1} = v_n e^{-2h} - u h e^{-2(h-\alpha h)} \nabla f\left(x_{n+\frac{1}{2}}\right) + 2\sqrt{u} \int_0^h e^{-2(h-s)}\, \mathrm{d}B_s,$$

where $x_{n+\frac{1}{2}}$ stands for the approximation of $x_n^*(\alpha h)$,

$$x_{n+\frac{1}{2}} = x_n + \frac{1 - e^{-2\alpha h}}{2} v_n - \frac{u}{2} \int_0^{\alpha h} \left(1 - e^{-2(\alpha h - s)}\right) \nabla f(x_n)\, \mathrm{d}s + \sqrt{u} \int_0^{\alpha h} \left(1 - e^{-2(\alpha h - s)}\right) \mathrm{d}B_s.$$

[6] See Appendix A.4

[7] The trajectory $x_n^*(s)$ refers to integral of the equation in ULD on $t \in [0, s]$: $\int_0^s dx_n^*(t) \ = \int_0^s v_n^*(t)\, ds$. Since the velocity is the derivative of position, we have $\int_0^s v_n^*(t) dt \ = x_n^*(s) - x_n$. However, $v_n^*(t)$ itself depends on the entire trajectory through $v_n^*(t) = v_n e^{-2t} - u \left(\int_0^t e^{-2(t-s)} \nabla f(x_n^*(s))\, ds \right) + 2\sqrt{u} \int_0^t e^{-2(t-s)}\, dB_s$. These two are mutually dependent and not computable.

[8] See Appendix A.6

These are the x_{n+1}, $x_{n+\frac{1}{2}}$ and v_{n+1} used in Algorithm 1.

Connection to collocation method As introduced in previous section, the collocation method views the exact solution as the operator $\mathcal{T}$: $x_n^* = \mathcal{T}(x_n^*) = x_n + \int_0^t F(x_n^*(s))ds$. RMM applies the representation with an unbiased randomized midpoint approximation $\mathbb{E}[h \cdot F(x_n^*(\alpha h))] = \int_0^h F(x_n^*(s))\, ds$. Repeating iterations of RMM can refine the approximation, minimizing the cumulative discretization error.

3 Theoretical Results

Assumptions The main assumption made to guarantee the algorithm performs is that the function f is a twice continuously differentiable function from $\mathbb{R}^d$ to $\mathbb{R}$ that has an L-Lipschitz continuous gradient and is m-strongly convex:

H1. *f is twice continuously differentiable on $\mathbb{R}^d$ and gradient L-Lipschitz: there exists $L \geq 0$ such that for all $x, y \in \mathbb{R}^d$,*

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

H2. *f is strongly convex, i.e. there exists $m > 0$ such that for all $x, y \in \mathbb{R}^d$,*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{m}{2}\|x - y\|^2.$$

The L-Lipschitz gradient assumption ensures that the gradient of $f(x)$ does not fluctuate too sharply over short distances. In practice, we cannot simulate a continuous-time diffusion process exactly and will have to approximate it through discretization. The smoothness of the gradient helps prevent errors accumulated over time.

Since a log-concave distribution is defined as one with a density of the form $\pi(x) \propto e^{-f(x)}$, any convex $f(x)$ leads to a log-concave target distribution $\pi(x)$. If $f(x)$ is also strongly convex, then $\pi(x)$ has a unique mode and decays rapidly in the tails, reducing the risk of long-tail behavior or multimodality. This structural property allows the sampling process to concentrate efficiently on the high-probability region within a limited number of steps, which plays crucial role in Main Theorem 3 that follows.

Unfortunately, in many non-convex optimization problems such as neural network training or posterior sampling in generative models, the function f is usually neither strongly convex nor smooth. In these cases, the theoretical guarantees of the algorithm may not hold.

Main Theorem Let f be a function such that $0 \prec m \cdot I_d \preceq \nabla^2 f(x) \preceq L \cdot I_d$ for all $x \in \mathbb{R}^d$. Let Y be a random point drawn from the density proportional to e^{-f}. Let the starting point x_0 be the point that minimizes $f(x)$ and $v_0 = 0$. For any $0 < \epsilon < 1$, if we set the step size of Algorithm 1 as $h = C \min(\frac{\epsilon^{1/3}}{\kappa^{1/6}} \log^{-1/6}(\frac{1}{\epsilon}), \epsilon^{2/3} \log^{-1/3}(\frac{1}{\epsilon}))$, for some small constant C and run the algorithm for $N = \frac{2\kappa}{h} \log(\frac{20}{\epsilon^2}) \leq \widetilde{\mathcal{O}}(\frac{\kappa^{7/6}}{\epsilon^{1/3}} + \frac{\kappa}{\epsilon^{2/3}})$ iterations, then Algorithm 1 after N iterations can generate a random point X such that $W_2(X, Y) \leq \epsilon\sqrt{\frac{d}{m}}$. Furthermore, each iteration of Algorithm 1 involves computing ∇f exactly twice.

Proof. In the randomized midpoint method (RMM), we denote by (x_n, v_n) the position and velocity at iteration n, and by $x_{n+1/2}$ an intermediate estimate of the position at a randomized time αh, where $\alpha \sim \text{Unif}[0, 1]$. To evaluate discretization error, we compare these iterates with the exact solution of the continuous-time underdamped Langevin diffusion (ULD), denoted (y_n, w_n), at time $t = nh$, where h is the step size. The exact ULD process is initialized at $(y_0, w_0) \sim \exp\left(-\left(f(y) + \frac{L}{2}\|w\|^2\right)\right)$, which is its stationary distribution. This ensures that the continuous-time process starts and remains in equilibrium, so any discrepancy between the algorithm and the exact process arises purely from discretization error rather than initialization bias.

Our goal is to show that the deviation between the algorithmic iterates (x_n, v_n) and the exact process (y_n, w_n) decays over time, up to a bounded discretization error. Specifically, our goal is to bound the quantity $\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right]$ after the N steps, so that it leads to an upper bound of the 2-Wasserstein distance $W_2(X, Y) = \mathbb{E}\left[\|x_N - y_N\|^2\right]$ between the distribution of the algorithm output and the target distribution.

First, we will use $\mathbb{E}_\alpha\left[\|x_n - y_n\|^2 + \|(x_n + v_n) - (y_n + w_n)\|^2\right]$ and use induction to derive the bound. By Proof Appendix B.1, we can show that:

$$\begin{aligned}\mathbb{E}_\alpha\left[\|x_n - y_n\|^2 + \|(x_n + v_n) - (y_n + w_n)\|^2\right] \leq & \left(1 + \frac{h}{2\kappa}\right)\left(\|y_n - x_n^*\|^2 + \|y_n + w_n - x_n^* - v_n^*\|^2\right) \\ & + \frac{2\kappa}{h}\left(\|\mathbb{E}_\alpha x_n - x_n^*\|^2 + \|\mathbb{E}_\alpha[x_n + v_n] - x_n^* - v_n^*\|^2\right) \\ & + \mathbb{E}_\alpha\|x_n + v_n - x_n^* - v_n^*\|^2 + \mathbb{E}_\alpha\|x_n - x_n^*\|^2\end{aligned}$$

where (x_n^*, v_n^*) is the one-step exact solution of the underdamped Langevin diffusion starting from (x_n, v_n). Now, we can apply this result to find the bound of $\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right]$. Please See Proof Appendix B.2 for more details in proof.

$$\begin{aligned}\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] &= \mathbb{E}\,\mathbb{E}_\alpha\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] \\ &\leq \left(1 + \frac{h}{2\kappa}\right)\mathbb{E}\left[\|y_N - x_N\|^2 + \|y_N + w_N - x_N - v_N\|^2\right] \\ &\quad + \frac{2\kappa}{h}\mathbb{E}\left[\|\mathbb{E}_\alpha x_N - x_N^*\|^2 + \|\mathbb{E}_\alpha[x_N + v_N] - x_N^* - v_N^*\|^2\right] \\ &\quad + \mathbb{E}\left[\|x_N - x_N^*\|^2 + \|x_N + v_N - x_N^* - v_N^*\|^2\right]\end{aligned}$$

Then, the idea is to use independence between each n steps and result at large N as the accumulation of the small step updates at each n (like the Telescopic Sum). Then we can write this as the following expression using recursion. (Please refer to the Proof Appendix B.2)

$$E_N \leq e^{-\frac{h}{2\kappa}} E_{N-1} + \delta_N \leq e^{-\frac{Nh}{2\kappa}} E_0 + \sum_{n=1}^{N} e^{-\frac{(N-n)h}{2\kappa}} \delta_n \leq e^{-\frac{Nh}{2\kappa}} E_0 + \delta_n$$

where,

- $E_N := \mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right]$
- $\delta_n := \frac{2\kappa}{h}\sum_{n=1}^{N}\left(2\mathbb{E}\left\|\mathbb{E}_\alpha v_n - v_n^*\right\|^2 + 3\mathbb{E}\left\|\mathbb{E}_\alpha x_n - x_n^*\right\|^2\right) + \sum_{n=1}^{N}\left(2\mathbb{E}\left\|v_n - v_n^*\right\|^2 + 3\mathbb{E}\left\|x_n - x_n^*\right\|^2\right)$

First, we want to bound E_0. To do that, We use **Proposition 1** (C.6) of [4], $\mathbb{E}\|y_0 - x_0\|^2 \leq \frac{d}{m}$. For limiting iteration times $N = \frac{2\kappa}{h}\log\left(\frac{20}{\epsilon^2}\right)$, the bound of E_0 becomes: (See Proof Appendix B.3)

$$e^{-\frac{Nh}{2\kappa}}\mathbb{E}\left[\|y_0 - x_0\|^2 + \|y_0 + w_0 - x_0 - v_0\|^2\right] = e^{-\frac{Nh}{2\kappa}} E_0 \leq \frac{\epsilon^2 d}{4m}.$$

Since we finished with bounding E_0, now we can bound δ_n using **Lemma 2** (C.3) (See Proof Appendix B.4)

$$\sum_{n=1}^{N}\frac{2\kappa}{h}\left(2\mathbb{E}\|\mathbb{E}_\alpha v_n - v_n^*\|^2 + 3\mathbb{E}\|\mathbb{E}_\alpha x_n - x_n^*\|^2\right) \leq O\left(h^7\kappa\sum_{n=0}^{N-1}\mathbb{E}\|v_n\|^2 + \frac{u}{m}h^9\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n)\|^2 + \frac{1}{m}Ndh^8\right)$$

$$\sum_{n=1}^{N} \left(2\mathbb{E}\|v_n - v_n^*\|^2 + 3\mathbb{E}\|x_n - x_n^*\|^2\right) \le O\left(h^4 \sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 + u^2 h^4 \sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 + Nudh^5\right)$$

By Proof Appendix B.5, when $N = \frac{2\kappa}{h}\log(\frac{20}{\epsilon^2})$, we know $\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2$ and $\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2$ are bounded by:

$$\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 \le O\left(\frac{\kappa dL}{h}\log\left(\frac{1}{\epsilon^2}\right) + \frac{L^2}{h} E_N\right)$$
$$\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 \le O\left(\frac{d}{hm}\log\left(\frac{1}{\epsilon^2}\right) + E_N\right)$$

Using these results, we can further bound δ_n when $N = \frac{2\kappa}{h}\log(\frac{20}{\epsilon^2})$:

$$\begin{aligned}
\delta_n &\le O\left(h^7\kappa \sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 + \frac{u}{m}h^9 \sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 + \frac{\kappa dh^7}{m}\log\left(\frac{1}{\epsilon^2}\right)\right) \\
&\quad + O\left(h^4 \sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 + u^2h^4 \sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 + 2\kappa udh^4 \log\left(\frac{1}{\epsilon^2}\right)\right) \quad \text{(plug in } N \text{ and ignore constants)} \\
&\le O\left(\left(\frac{\kappa dh^6}{m} + \frac{dh^3}{m}\right)\log\left(\frac{1}{\epsilon^2}\right)\right) + O\left(\kappa h^7 + h^3\right) E_N \quad \text{(keep only } \|v_n\|^2 \text{ terms).}
\end{aligned}$$

By Proof Appendix B.6 and B.7, we know that $O\left(\left(\frac{\kappa dh^6}{m} + \frac{dh^3}{m}\right)\log\left(\frac{1}{\epsilon^2}\right)\right) \le \frac{\epsilon^2 d}{4m}$ and $O\left(\kappa h^7 + h^3\right) E_N \le \frac{1}{2}E_N$ for some chosen $h = C\min\left(\frac{\epsilon^{1/3}}{\kappa^{1/6}}\log^{-1/6}\left(\frac{1}{\epsilon^2}\right),\ \epsilon^{2/3}\log^{-1/3}\left(\frac{1}{\epsilon^2}\right)\right)$. Finally we can bound δ_n:

$$\delta_n \le \frac{\epsilon^2 d}{4m} + \frac{1}{2}E_N.$$

Therefore, combining all of our final bounds, we can get a final bound of E_N:

$$E_N \le e^{-\frac{Nh}{2\kappa}} E_0 + \sum_{n=1}^{N} \delta_n \le \frac{\epsilon^2 d}{4m} + \frac{\epsilon^2 d}{4m} + \frac{1}{2}E_N = \frac{\epsilon^2 d}{2m} + \frac{1}{2}E_N \Rightarrow E_N \le \frac{\epsilon^2 d}{m}$$

Therefore, for $h = C\min\left(\frac{\epsilon^{1/3}}{\kappa^{1/6}}\log^{-1/6}\left(\frac{1}{\epsilon^2}\right),\ \epsilon^{2/3}\log^{-1/3}\left(\frac{1}{\epsilon^2}\right)\right)$, we can show that the Wasserstein distance is bounded by:

$$W_2^2(X,Y) = \mathbb{E}\left[\|x_N - y_N\|^2\right] \le \mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] = E_N \le \frac{\epsilon^2 d}{m}$$

$$\text{(Plug in } h\text{)}\ N = \frac{2\kappa}{h}\log(\frac{20}{\epsilon^2}) \sim \max\left(\frac{\kappa^{7/6}}{\epsilon^{1/3}}, \frac{\kappa}{\epsilon^{2/3}}\right) \le \tilde{O}(\frac{\kappa^{7/6}}{\epsilon^{1/3}} + \frac{\kappa}{\epsilon^{2/3}})$$

□

A Appendix

A.1 Complexity of previous work

Table 1: Summary of iteration complexity. Except for Hit-and-Run, each step involves $O(1)$ gradient computation. Hit-and-Run uses $O(1)$ function value computations per step.

Algorithm	**# Step**	
	Warm Start	**Cold Start**
Hit-and-Run	$\widetilde{O}(d^3 \log(1/\epsilon))$	$\widetilde{O}(d^4 \log(1/\epsilon))$
Langevin Diffusion	$\widetilde{O}(\kappa^2/\epsilon^2)$	
Underdamped Langevin Diffusion	$\widetilde{O}(\kappa^2/\epsilon)$	
Underdamped Langevin Diffusion 2	$\widetilde{O}(\kappa^{1.5}/\epsilon + \kappa^2)$	
High-Order Langevin Diffusion	$\widetilde{O}(\kappa^{19/4}/\epsilon^{1/2} + \kappa^{13/3}/\epsilon^{2/3})$	
Metropolis-Adjusted Langevin Algorithm	$\widetilde{O}((\kappa d + \kappa^{1.5}\sqrt{d}) \log(1/\epsilon))$	$\widetilde{O}((\kappa d^2 + \kappa^{1.5} d^{1.5}) \log(1/\epsilon))$
Hamiltonian Monte Carlo with Euler Method	$\widetilde{O}(\kappa^{6.5}/\epsilon)$	
Hamiltonian Monte Carlo with Collocation Method	$\widetilde{O}(\kappa^{1.75}/\epsilon)$	
Hamiltonian Monte Carlo with Collocation Method 2	$\widetilde{O}(\kappa^{1.5}/\epsilon)$	
Underdamped Langevin Diffusion with Randomized Midpoint Method (This Paper)	$\widetilde{O}(\kappa^{7/6}/\epsilon^{1/3} + \kappa/\epsilon^{2/3})$	

As shown, the underdamped Langevin diffusion (ULD) combined with the randomized midpoint method (RMM) achieves a convergence rate that is independent of the dimension d. This is particularly advantageous when working with high-dimensional datasets. Moreover, the number of steps required exhibits only mild dependence on the condition number κ, which quantifies how ill-conditioned the target distribution is. Algorithms with strong dependence on κ tend to perform poorly when sampling from sharply curved or anisotropic distributions. Lastly, the step complexity has reduced dependence on $1/\epsilon$, which is critical when high-accuracy solutions are required, as a strong dependence would lead to excessive computational cost.

A.2 Wasserstein distance

The Wasserstein distance is a way to measure how similar or different two probability distributions are, taking into account the geometry of the underlying space. Intuitively, it represents the minimal “cost” of transporting mass to one distribution into another, where the cost is defined by how far the mass has to move. For two probability measures μ and ν on $\mathbb{R}^d$, the 2-Wasserstein distance is

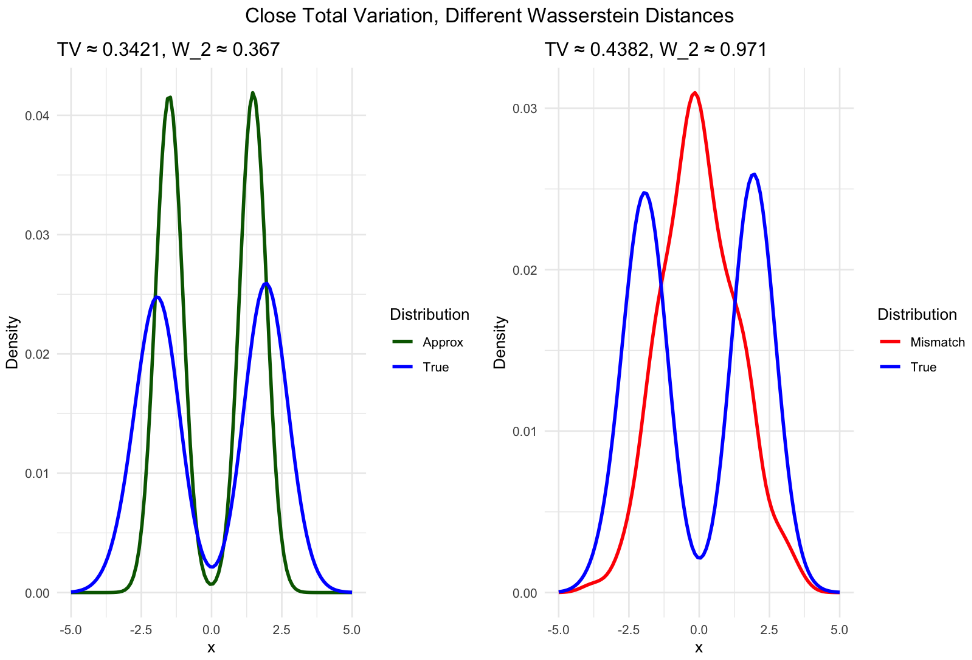

Figure 1: Scenario where 2-Wasserstein excels due to geometry-awareness

defined as:

$$W_2(\mu,\nu)^2 = \inf_{\gamma\in\Gamma(\mu,\nu)} \int \|x-y\|^2 \, d\gamma(x,y),$$

where $\Gamma(\mu,\nu)$ is the set of all couplings (joint distributions) with marginals μ and ν. In this paper, we measure sampling error in terms of W_2. Our main result shows that, after

$$N \;=\; \widetilde{O}\Big(\kappa^{7/6}\,\varepsilon^{-1/3} \;+\; \kappa\,\varepsilon^{-2/3}\Big)$$

steps of RMM, the 2-Wasserstein distance between the discrete output and the true target is at most $\varepsilon\sqrt{d/m}$.

The Wasserstein distance is geometry-aware and coupling-friendly, making it an ideal measure in assessing sampler performance. Figure 1 demonstrates a simple case in which Wasserstein distance is more effective than other measures: While a metric like total variation distribution difference metric like total variation shows similar values for these drastically different simulated distributions, Wasserstein distance more accurately captures the differences due to its characteristics.

A.2.1 Comparing W_2 against Total Variance

- **Geometry-awareness.** Total variation distance

$$\mathrm{TV}(\mu,\nu) \;=\; \sup_{A\subset\mathbb{R}^d} \big|\mu(A)-\nu(A)\big|$$

measures only the largest pointwise discrepancy in probability mass, without regard to how “far apart” those mismatched bits of mass lie in $\mathbb{R}^d$. In contrast, the 2-Wasserstein distance

$$W_2(\mu,\nu) \;=\; \inf_{\gamma\in\Gamma(\mu,\nu)} \Big(\mathbb{E}_{(X,Y)\sim\gamma}\big[\|X-Y\|^2\big]\Big)^{1/2}$$

explicitly penalizes the Euclidean distance $\|X - Y\|$ needed to "transport" mass from μ to ν. As a result, W_2 remains small whenever μ and ν differ only by a slight shift or small deformation—even if their densities have disjoint support—whereas $\mathrm{TV}(\mu, \nu)$ can be arbitrarily large (up to 1) in that same situation. In sampling, discretization errors typically manifest as small local shifts in the continuous-time trajectory; W_2 captures these gradual discrepancies, while TV cannot distinguish between a tiny shift and a completely orthogonal distribution.

- **Coupling-friendliness.** The Langevin dynamics (both overdamped and underdamped) contract exponentially in W_2 (see Lemma C.2). Contraction in total variation would require much stronger functional-inequality assumptions and even then only controls TV up to a constant. Since the randomized midpoint method couples each discrete step to the exact SDE via a shared Brownian motion, the error analysis relies on estimating $\mathbb{E}\big[\|x_n - y_n\|^2\big]$ under that coupling. This is exactly the squared W_2–distance along a particular coupling, so we obtain a clean per-step "contraction + discretization-error" decomposition. No comparably simple per-step estimate exists for TV.

- **Sensitivity to small perturbations.** If two distributions differ by only an $\eta\%$ change in density within a ball of radius $\delta \ll 1$, then

$$\mathrm{TV}(\mu, \nu) \;\approx\; \eta, \quad W_2(\mu, \nu) \;\approx\; \eta\,\delta.$$

  Thus, as $\delta \to 0$, $W_2 \to 0$ even if TV stays at η. In discretizing a continuous trajectory, the local bias is typically of order $O(h^2)$ or $O(h^3)$ in Euclidean norm; since W_2 "sees" that small shift, the overall convergence rate reflects those high-order terms. In contrast, TV often remains $O(1)$ until the approximation is uniformly close to the target density, forcing very small step-sizes and yielding weaker complexity bounds.

- **Practical implication.** Because W_2 is finite for log-concave targets on $\mathbb{R}^d$ (so long as second moments exist) and contracts under ULD, we can prove dimension-free convergence rates $\widetilde{O}\big(\kappa^{7/6}\,\varepsilon^{-1/3} + \kappa\,\varepsilon^{-2/3}\big)$. Any analogous bound in total variation would either blow up with d or incur an ε^{-2} dependence, drastically worsening the complexity guarantee.

A.3 Stationary distribution of ULD

ULD represents the stochastic dynamics of a particle and we can model the probability of finding the particle at position x, velocity v, at time t as probability density $p(x, v, t)$. In order to find the stationary distribution of the ULD, we introduce the *Fokker-Planck Equation*:

$$\frac{\partial p}{\partial t} = \text{probability of flowing in} - \text{probability of flowing out}$$

to describe how the probability density flows over time.

If ULD runs for a long enough time, the flows of probability balance perfectly which in Fokker-Planck Equation is

$$\frac{\partial p}{\partial t} = 0$$

where we have the exact solution:

$$\exp\left(-(f(x) + \frac{1}{2u}\|v\|^2)\right)$$

Since we set $u = \frac{1}{L}$, the stationary ditribution of ULD is:

$$p(x, v) \sim \exp\left(-(f(x) + \frac{L}{2}\|v\|^2)\right)$$

Then take the marginal distribution of v, we have

$$p(v) \sim \exp\left(-\frac{L}{2}\|v\|^2\right) \Rightarrow v \sim \mathcal{N}\left(0, \frac{1}{L}I\right)$$

Hence, the expectation of v^2 can be computed as

$$\begin{gathered}\sqrt{L}v \sim \mathcal{N}(0, I)\\ L\|v\|^2 \sim \chi^2(d)\\ \mathbb{E}L\|v\|^2 = d\\ \mathbb{E}\|v\|^2 = \frac{d}{L}\end{gathered}$$

A.4 Integral Formulation of ULD (with RMM)

As mentioned above, the paper notes that the ULD follows the SDE:

$$\begin{aligned}dv(t) &= -2v(t)\,dt - u\nabla f(x(t))\,dt + 2\sqrt{u}\,dB_t,\\ dx(t) &= v(t)\,dt,\end{aligned}$$

where $u = \frac{1}{L}$.

A core part of the paper's construction and contribution relies on studying the integral formulation of this:

$$\begin{aligned}x_n^*(t) &= x_n + \frac{1-e^{-2t}}{2}v_n - \frac{u}{2}\int_0^t \left(1 - e^{-2(t-s)}\right)\nabla f(x_n^*(s))\,\mathrm{d}s + \sqrt{u}\int_0^t \left(1 - e^{-2(t-s)}\right)\mathrm{d}B_s,\\ v_n^*(t) &= v_n e^{-2t} - u\left(\int_0^t e^{-2(t-s)}\nabla f(x_n^*(s))\,\mathrm{d}s\right) + 2\sqrt{u}\int_0^t e^{-2(t-s)}\,\mathrm{d}B_s.\end{aligned}$$

and that according to [1] we can use $\nabla f(x_n)$ to approximate $\nabla f(x_n^*(t)$ for $t \in [0, h]$ to get the following algorithm:

$$\begin{aligned}\hat{x}_n(h) &= x_n + \frac{1-e^{-2h}}{2}v_n - \frac{u}{2}\int_0^h \left(1 - e^{-2(h-s)}\right)\nabla f(x_n)\,\mathrm{d}s + \sqrt{u}\int_0^h \left(1 - e^{-2(h-s)}\right)\mathrm{d}B_s,\\ \hat{v}_n(h) &= v_n e^{-2h} - u\left(\int_0^h e^{-2(h-s)}\nabla f(x_n)\,\mathrm{d}s\right) + 2\sqrt{u}\int_0^h e^{-2(h-s)}\,\mathrm{d}B_s.\end{aligned}$$

The proposed RMM helps to identify an accurate estimator of the integral $\int_0^h (1 - e^{-2(h-s)})\nabla f(x_n^*(s))ds$, and in doing so we work with the following:

$$\begin{aligned}x_{n+\frac{1}{2}} &= x_n + \frac{1-e^{-2\alpha h}}{2}v_n - \frac{u}{2}\int_0^{\alpha h} \left(1 - e^{-2(\alpha h-s)}\right)\nabla f(x_n)\,\mathrm{d}s + \sqrt{u}\int_0^{\alpha h} \left(1 - e^{-2(\alpha h-s)}\right)\mathrm{d}B_s\\ x_{n+1} &= x_n + \frac{1-e^{-2h}}{2}v_n - \frac{u}{2}h\left(1 - e^{-2(h-\alpha h)}\right)\nabla f\left(x_{n+\frac{1}{2}}\right) + \sqrt{u}\int_0^h \left(1 - e^{-2(h-s)}\right)\mathrm{d}B_s,\\ v_{n+1} &= v_n e^{-2h} - uhe^{-2(h-\alpha h)}\nabla f\left(x_{n+\frac{1}{2}}\right) + 2\sqrt{u}\int_0^h e^{-2(h-s)}\,\mathrm{d}B_s,\end{aligned}$$

Appendix A.5 details the implementation of W_1, W_2, W_3 that covers the randomness. This appendix will cover how we go from the SDE to the integral formulation to begin with.

A.4.1 Deriving the Integral Formulation of ULD

We start from the velocity function in ULD system:

$$dv(t) = -2v(t)\,dt - u\nabla f(x(t))\,dt + 2\sqrt{u}\,dB_t.$$

Multiply e^{2t} on both sides and move $-2e^{2t}v(t)\,dt$ to left hand side:

$$e^{2t}dv(t) + 2e^{2t}v(t)\,dt = -ue^{2t}\nabla f(x(t))\,dt + 2\sqrt{u}e^{2t}\,dB_t.$$

Notice that the left hand side is the differential of $e^{2t}v(t)$:

$$d(e^{2t}v(t)) = -ue^{2t}\nabla f(x(t))\,dt + 2\sqrt{u}e^{2t}\,dB_t.$$

Then we take integral on $[0,t]$ for both sides:

$$e^{2t}v_n(t) - e^0 v_n = \int_0^t -ue^{2s}\nabla f(x(s))\,ds + \int_0^t 2\sqrt{u}e^{2s}\,dB_s.$$

Divide e^{2t} on both sides and move v_n to the right hand side, we will have the exact solution to $v_n^*(t)$:

$$v_n^*(t) = \frac{v_n}{e^{2t}} - u\int_0^t e^{-2(t-s)}\nabla f(x(s))\,ds + 2\sqrt{u}\int_0^t e^{-2(t-s)}\,dB_s.$$

For $x_n^*(t)$, we take integral to position function in ULD system on $[0,t]$ first:

$$dx(t) = v(t)\,dt \Rightarrow \int_0^t dx(s) = \int_0^t v(s)\,ds \Rightarrow x(t) = x_n + \int_0^t v(s)\,ds,$$

where $v(s)$ here is the exact solution $v_n^*(t)$. Replace $v(s)$ with $v_n^*(s)$:

$$x(t) = x_n + \int_0^t [v_n e^{-2s} - u\int_0^s e^{-2(s-r)}\nabla f(x(r))\,dr + 2\sqrt{u}\int_0^s e^{-2(s-r)}\,dB_r]ds.$$

We solve these integrations separately:

- $$\int_0^t v_n e^{-2s}\,ds = v_n\int_0^t e^{-2s}\,ds = \frac{1-e^{-2t}}{2}v_n.$$

- $$\int_0^t (\int_0^s e^{-2(s-r)}\nabla f(x(r))\,dr)ds = \int_0^t (\int_r^t e^{-2(s-r)}\,ds)\nabla f(x(r))\,dr,$$

where $0 \le r \le s \le t$.
Let $\tau = s - r$,

$$\int_r^t e^{-2(s-r)}\,ds = \int_0^{t-r} e^{-2\tau}\,d\tau = \frac{1-e^{-2(t-r)}}{2}.$$

Then we have

$$-u\int_0^t (\int_r^t e^{-2(s-r)}\,ds)\nabla f(x(r))\,dr = -\frac{u}{2}\int_0^t (1-e^{-2(t-r)})\nabla f(x(r))\,dr = -\frac{u}{2}\int_0^t (1-e^{-2(t-s)})\nabla f(x(s))\,ds.$$

With the similar idea, we can solve the third part:

$$\int_0^t 2\sqrt{u}\int_0^s e^{-2(s-r)}\,dB_r\,ds \Rightarrow 2\sqrt{u}\int_0^t (\int_r^t e^{-2(s-r)}\,ds)dB_r,$$

where $0 \leq r \leq s \leq t$.

$$-u \int_0^t (\int_r^t e^{-2(s-r)}\, ds) \nabla f(x(r))\, dr = \sqrt{u} \int_0^t (1 - e^{-2(t-r)})\, dB_r = \sqrt{u} \int_0^t (1 - e^{-2(t-s)})\, dB_s.$$

Put the three parts together, we can get the exact solution $x_n^*(t)$:

$$x_n^*(t) = x_n + \frac{1 - e^{-2t}}{2} v_n - \frac{u}{2} \int_0^t (1 - e^{-2(t-s)}) \nabla f(x(s))\, ds + \sqrt{u} \int_0^t (1 - e^{-2(t-s)})\, dB_s$$

A.4.2 Intuition for Integral Formulation of ULD

This integral formulation reveals a key idea: the stochastic system *forgets the past exponentially.* In other words:

- Past values of $v(t)$ and noise have have a *diminishing impact* on the current state due to damping (the term $-2v(t)$.
- This fading memory shows up as the exponential decay factor $e^{-2(t-s)}$ in the integrals.
- For position, the velocity is integrated over time, so we see a "build-up" weight $1 - e^{-2(t-s)}$, which starts at 0 and grows to 1. This reflects how noise gradually affects position.
- The noise terms in the SDE are not raw Brownian motion, but filtered through these exponential weights, which give rise to the specifically weighted integrals W_1, W_2, W_3 detailed in Appendix A.5.

This ties directly into how we simulate noise in RMM: we are replicating the effect of Brownian motion as it behaves under the physics of the damped stochastic system, rather than injecting generic Gaussian noise.

A.5 Brownian Motion Simulation

A key component of the randomized midpoint method is the simulation of Brownian motion terms that arise from discretizing the underdamped Langevin diffusion (ULD). Since ULD is a stochastic differential equation, it includes random noise components modeled as integrals with respect to Brownian motion. These integrals must be accurately simulated to ensure that the discrete-time algorithm closely matches the behavior of the continuous process.

In Algorithm 1, the updates for position and velocity involve three random vectors:

$$\begin{aligned} W_1 &= \int_0^{\alpha h} (1 - e^{-2(\alpha h - s)}) dB_s, \\ W_2 &= \int_0^{h} (1 - e^{-2(h - s)}) dB_s, \\ W_3 &= \int_0^{h} e^{-2(h - s)} dB_s. \end{aligned}$$

Specifically, if current step size is h, and we choose a random midpoint $\alpha h \in [0, h]$. Then W_1 is used to simulate noise when computing midpoint $x_{n+\frac{1}{2}}$, W_2 is used in the position update x_{n+1}, and W_3 is used in the velocity update v_{n+1}. W_1, W_2, W_3 are all vector-valued Gaussian random variables because they are integrals over time with specific weights. These integrals are not standard Gaussian variables, but they are still Gaussian with mean zero and computable covariances. Therefore, the task becomes how to efficiently sample these specific Gaussian variables. We also note that the W_2 used as part of the algorithm is not the same as W_2 used for Wasserstein distance.

To do this, we break the time interval $[0, h]$ into two segments: $[0, \alpha h]$ and $[\alpha h, h]$, where $\alpha \in [0, 1]$ is chosen uniformly at random each iteration. We then define four auxiliary random variables:

$$G_1 = \int_0^{\alpha h} e^{2s}, dB_s, \qquad H_1 = \int_0^{\alpha h} dB_s,$$
$$G_2 = \int_{\alpha h}^{h} e^{2s}, dB_s, \qquad H_2 = \int_{\alpha h}^{h} dB_s.$$

We utilize these components to express the Brownian integrals as:

$$\begin{aligned} W_1 &= H_1 - e^{-2\alpha h} G_1, \\ W_2 &= (H_1 + H_2) - e^{-2h}(G_1 + G_2), \\ W_3 &= e^{-2h}(G_1 + G_2). \end{aligned}$$

Because each of G_1, H_1, G_2, H_2 are Gaussian vectors, and the two pairs (G_1, H_1) and (G_2, H_2) are independent (they are over disjoint time intervals), it is sufficient to sample these four Gaussian vectors correctly.

A.5.1 Intuition for Weighted Brownian Integrals W_1, W_2, W_3

A natural question that arises is: *Why not just use standard Gaussian noise instead of simulating these specific integrals?*

The reason lies in the structure of the integrals themselves. The weighted Brownian integrals W_1, W_2, W_3 are designed to match the statistical behavior of the continuous-time ULD process over short intervals. Using standard Gaussian noise would ignore the time-weighted accumulation and decay of randomness dictated by the process dynamics. This would introduce systematic bias into the simulation and degrade the convergence guarantees. By preserving the exact covariance structure through correct simulation, we ensure the discretization mirrors the underlying SDE and avoids needing Metropolis corrections to fix bias.

Another question that is helpful to answer is: *Why the weighted Brownian integrals have different weights?*

The short answer is: the different weightings arise from solving the underdamped Langevin diffusion (ULD) in integral form, where exponential decay naturally appears due to the damping term in the SDE (covered in Appendix A.4). Specifically:

- The velocity update equation includes a term $-2v(t)$, which leads to exponential decay of past velocity contributions. Solving this differential equation gives rise to expressions involving $e^{-2(t-s)}$ as a time-decay kernel.

- $W_3 = \int_0^h e^{-2(h-s)} dB_s$ directly comes from this exponential decay in the velocity's Brownian component. Earlier noise (small s) contributes less due to damping.

- $W_2 = \int_0^h (1 - e^{-2(h-s)}) dB_s$ comes from integrating the velocity over time to get position. Because $dx(t) = v(t)dt$, the total effect of Brownian noise on position accumulates with a weighting that reflects how much velocity was affected over time. This leads to a cumulative "build-up" form: $1 - e^{-2(h-s)}$.

- $W_1 = \int_0^{\alpha h} (1 - e^{-2(\alpha h - s)}) dB_s$ is just the same structure as W_2, but up to the random midpoint αh. It captures the random midpoint's noise impact, which is used to estimate the gradient more accurately at a representative time.

So in summary:

- $e^{-2(h-s)}$ in W_3: represents exponential *decay* of velocity noise due to damping.
- $1 - e^{-2(h-s)}$ in W_2: represents *cumulative effect* of velocity noise on position.
- $1 - e^{-2(\alpha h-s)}$ in W_1: same as W_2, but on a subinterval for midpoint approximation.

Each weighting reflects how noise impacts different components of the dynamics: W_3 models immediate velocity damping, W_2 integrates over velocity's effect on position, and W_1 is needed to compute an unbiased drift estimate using a randomized collocation.

A.5.2 Simulation Details from Original Paper

The following lemma in the paper demonstrates how to simulate the Brownian motion terms W_1, W_2, W_3 used in the randomized midpoint method (correctly and efficiently).

Lemma 5. Define $G_1 = \int_0^{\alpha h} e^{2s}\, dB_s$, $G_2 = \int_{\alpha h}^{h} e^{2s}\, dB_s$, $H_1 = \int_0^{\alpha h} dB_s$ and $H_2 = \int_{\alpha h}^{h} dB_s$. Then, (G_1, H_1) is independent of (G_2, H_2). Moreover, (G_1, H_1) and (G_2, H_2) both follow a 2d-dimensional Gaussian distribution with mean zero. *Conditional on the choice of α, their covariance is given by*

$$
\begin{aligned}
\mathbb{E}\left[(G_1 - \mathbb{E}G_1)(H_1 - \mathbb{E}H_1)^T\right] &= \frac{1}{2}\left(e^{2\alpha h} - 1\right) \cdot I_d, \\
\mathbb{E}\left[(G_1 - \mathbb{E}G_1)(G_1 - \mathbb{E}G_1)^T\right] &= \frac{1}{4}\left(e^{4\alpha h} - 1\right) \cdot I_d, \\
\mathbb{E}\left[(H_1 - \mathbb{E}H_1)(H_1 - \mathbb{E}H_1)^T\right] &= \alpha h \cdot I_d, \\
\mathbb{E}\left[(G_2 - \mathbb{E}G_2)(H_2 - \mathbb{E}H_2)^T\right] &= \frac{1}{2}\left(e^{2h} - e^{2\alpha h}\right) \cdot I_d, \\
\mathbb{E}\left[(G_2 - \mathbb{E}G_2)(G_2 - \mathbb{E}G_2)^T\right] &= \frac{1}{4}\left(e^{4h} - e^{4\alpha h}\right) \cdot I_d, \\
\mathbb{E}\left[(H_2 - \mathbb{E}H_2)(H_2 - \mathbb{E}H_2)^T\right] &= (h - \alpha h) \cdot I_d.
\end{aligned}
$$

Proof. By the definition of the standard Brownian motion, (G_1, H_1) is independent of (G_2, H_2) and both (G_1, H_1) and (G_2, H_2) have mean zero. Moreover,

$$
\begin{aligned}
\mathbb{E}\left[(G_1 - \mathbb{E}G_1)(H_1 - \mathbb{E}H_1)^T\right] &= \mathbb{E}\left[\left(\int_0^{\alpha h} e^{2s}\, dB_s\right)\left(\int_0^{\alpha h} dB_s\right)^T\right] \\
&= \int_0^{\alpha h} e^{2s}\, ds \cdot I_d \\
&= \frac{1}{2}\left(e^{2\alpha h} - 1\right) \cdot I_d,
\end{aligned}
$$

$$\begin{aligned}\mathbb{E}\left[(G_1 - \mathbb{E}G_1)(G_1 - \mathbb{E}G_1)^T\right] &= \mathbb{E}\left[\left(\int_0^{\alpha h} e^{2s}\, dB_s\right)\left(\int_0^{\alpha h} e^{2s}\, dB_s\right)^T\right] \\ &= \int_0^{\alpha h} e^{4s}\, ds \cdot I_d \\ &= \frac{1}{4}\left(e^{4\alpha h} - 1\right) \cdot I_d,\end{aligned}$$

and

$$\mathbb{E}\left[(H_1 - \mathbb{E}H_1)(H_1 - \mathbb{E}H_1)^T\right] = \alpha h \cdot I_d.$$

Similarly,

$$\begin{aligned}\mathbb{E}\left[(G_2 - \mathbb{E}G_2)(H_2 - \mathbb{E}H_2)^T\right] &= \mathbb{E}\left[\left(\int_{\alpha h}^{h} e^{2s}\, dB_s\right)\left(\int_{\alpha h}^{h} dB_s\right)^T\right] \\ &= \int_{\alpha h}^{h} e^{2s}\, ds \cdot I_d \\ &= \frac{1}{2}\left(e^{2h} - e^{2\alpha h}\right) \cdot I_d,\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[(G_2 - \mathbb{E}G_2)(G_2 - \mathbb{E}G_2)^T\right] &= \mathbb{E}\left[\left(\int_{\alpha h}^{h} e^{2s}\, dB_s\right)\left(\int_{\alpha h}^{h} e^{2s}\, dB_s\right)^T\right] \\ &= \int_{\alpha h}^{h} e^{4s}\, ds \cdot I_d \\ &= \frac{1}{4}\left(e^{4h} - e^{4\alpha h}\right) \cdot I_d,\end{aligned}$$

and

$$\mathbb{E}\left[(H_2 - \mathbb{E}H_2)(H_2 - \mathbb{E}H_2)^T\right] = (h - \alpha h) \cdot I_d.$$

□

A.6 Unbiasedness of Randomized Midpoint Method

In numerical discretization of SDEs, a method is said to be *unbiased* if the expected value of the numerical update matches the infinitesimal generator of the SDE, at least to leading order. For example, the Euler–Maruyama method evaluates the drift at the left endpoint x_n, leading to the discretization

$$x_{n+1} = x_n - h\nabla f(x_n) + \sqrt{2h}\,\xi_n,$$

which introduces bias because it systematically under- or overestimates the average drift over the interval $[t_n, t_{n+1}]$. Specifically, since $\nabla f(x(t))$ typically varies over time as $x(t)$ evolves, evaluating it at the left endpoint x_n ignores this curvature. If the gradient increases over the interval (i.e., the function is locally convex), Euler underestimates the total drift. If the gradient decreases (as in concave regions), it overestimates it. In both cases, this mismatch accumulates over multiple steps and leads to a biased trajectory that diverges from the true behavior of the SDE.

In contrast, the *Randomized Midpoint Method* (RMM) selects a random interpolation point $\alpha \sim \text{Uniform}(0,1)$ and evaluates the drift at $x_n + \alpha h v_n$. The update takes the form:

$$x_{n+1} = x_n - h\nabla f(x_n + \alpha h v_n) + \sqrt{2h}\,\xi_n.$$

Taking expectation with respect to α, we obtain:

$$\mathbb{E}_\alpha\left[\nabla f(x_n + \alpha h v_n)\right] = \frac{1}{h}\int_0^h \nabla f(x_n + s v_n)\,ds.$$

This is a Monte Carlo estimate of the average drift over the interval $[x_n, x_n + hv_n]$. As a result, the RMM provides an unbiased estimator of the integrated drift term. In expectation, this matches the continuous-time behavior of the SDE, and therefore removes the deterministic bias present in fixed-point evaluations like Euler's method.

B Proof Appendix

This section details the proof of the main theorem (3).

B.1 Bound of $\mathbb{E}\left[\|x_n^* - y_n\|^2 + \|(x_n^* + v_n^*) - (y_n + w_n)\|^2\right]$

To begin, we insert the one-step exact solution (x_n^*, v_n^*) of the underdamped Langevin diffusion starting from (x_n, v_n), and decompose the total error using simple norm calculation and linearity of expectation:

$$\begin{aligned}
&\mathbb{E}_\alpha\left[\|x_n - y_n\|^2 + \|(x_n + v_n) - (y_n + w_n)\|^2\right] \\
&\qquad = \mathbb{E}_\alpha\left[\|(x_n - x_n^*) - (y_n - x_n^*)\|^2 + \|(x_n + v_n - x_n^* - v_n^*) - (y_n + w_n - x_n^* - v_n^*)\|^2\right] \\
&\qquad = \|y_n - x_n^*\|^2 + \|y_n + w_n - x_n^* - v_n^*\|^2 \\
&\qquad\quad + \mathbb{E}_\alpha\|x_n - x_n^*\|^2 + \mathbb{E}_\alpha\|x_n + v_n - x_n^* - v_n^*\|^2 \\
&\qquad\quad - 2(y_n - x_n^*)^\top(\mathbb{E}_\alpha x_n - x_n^*) \\
&\qquad\quad - 2(y_n + w_n - x_n^* - v_n^*)^\top(\mathbb{E}_\alpha[x_n + v_n] - x_n^* - v_n^*)
\end{aligned}$$

Then, using Young's inequality: $-2a^\top b \leq \frac{1}{\varepsilon}\|a\|^2 + \varepsilon\|b\|^2$, we have

$$-2(y_n - x_n^*)^T\mathbb{E}_\alpha[x_n - x_n^*] \leq \frac{h}{2\kappa}\|y_n - x_n^*\|^2 + \frac{2\kappa}{h}\|\mathbb{E}_\alpha[x_n - x_n^*]\|^2,$$

$$-2(y_n + w_n - x_n^* - v_n^*)^T\mathbb{E}_\alpha[x_n + v_n - x_n^* - v_n^*] \leq \frac{h}{2\kappa}\|y_n + w_n - x_n^* - v_n^*\|^2 + \frac{2\kappa}{h}\|\mathbb{E}_\alpha[x_n + v_n - x_n^* - v_n^*]\|^2.$$

Hence, we can bound the previous results as:

$$\begin{aligned}
&\|y_n - x_n^*\|^2 + \|y_n + w_n - x_n^* - v_n^*\|^2 + \mathbb{E}_\alpha\|x_n - x_n^*\|^2 + \mathbb{E}_\alpha\|x_n + v_n - x_n^* - v_n^*\|^2 \\
&\quad - 2(y_n - x_n^*)^T\mathbb{E}_\alpha[x_n - x_n^*] - 2(y_n + w_n - x_n^* - v_n^*)^T\mathbb{E}_\alpha[x_n + v_n - x_n^* - v_n^*] \\
&\leq \left(1 + \frac{h}{2\kappa}\right)\left(\|y_n - x_n^*\|^2 + \|y_n + w_n - x_n^* - v_n^*\|^2\right) \\
&\quad + \frac{2\kappa}{h}\left(\|\mathbb{E}_\alpha x_n - x_n^*\|^2 + \|\mathbb{E}_\alpha[x_n + v_n] - x_n^* - v_n^*\|^2\right) + \mathbb{E}_\alpha\|x_n + v_n - x_n^* - v_n^*\|^2 + \mathbb{E}_\alpha\|x_n - x_n^*\|^2
\end{aligned}$$

B.2 Expansion and Bound of $\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right]$

$$\begin{aligned}
&\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] \\
=&\mathbb{E}\mathbb{E}_\alpha\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] \\
=&\mathbb{E}\mathbb{E}_\alpha\left[\|(x_N - x_N^*) - (y_N - x_N^*)\|^2 + \|(x_N + v_N - x_N^* - v_N^*) - (y_N + w_N - x_N^* - v_N^*)\|^2\right] \\
=&\mathbb{E}\Big[\|y_N - x_N^*\|^2 + \|y_N + w_N - x_N^* - v_N^*\|^2 + \mathbb{E}_\alpha\|x_N - x_N^*\|^2 + \mathbb{E}_\alpha\|x_N + v_N - x_N^* - v_N^*\|^2 \\
&-2(y_N - x_N^*)^\top(\mathbb{E}_\alpha x_N - x_N^*) - 2(y_N + w_N - x_N^* - v_N^*)^\top(\mathbb{E}_\alpha x_N + v_N - x_N^* - v_N^*)\Big]
\end{aligned}$$

By using the result from Proof Appendix B.1

$$\begin{aligned}
\leq &\left(1 + \frac{h}{2\kappa}\right)\mathbb{E}\left[\|y_N - x_N\|^2 + \|y_N + w_N - x_N - v_N\|^2\right] \\
&+ \frac{2\kappa}{h}\left(\mathbb{E}\|\mathbb{E}_\alpha x_N - x_N^*\|^2 + \mathbb{E}\|\mathbb{E}_\alpha x_N + v_N - x_N^* - v_N^*\|^2\right) \\
&+ \left(\mathbb{E}\|x_N - x_N^*\|^2 + \mathbb{E}\|x_N + v_N - x_N^* - v_N^*\|^2\right)
\end{aligned}$$

and by **Lemma 1** (C.2) and set $t = h$,

$$\begin{aligned}
&\left(1 + \frac{h}{2\kappa}\right)\mathbb{E}\left[\|y_N - x_N\|^2 + \|y_N + w_N - x_N - v_N\|^2\right] \\
\leq &\left(1 + \frac{h}{2\kappa}\right)e^{-\frac{h}{\kappa}}\mathbb{E}\left[\|y_{N-1} - x_{N-1}\|^2 + \|y_{N-1} + w_{N-1} - x_{N-1} - v_{N-1}\|^2\right]
\end{aligned}$$

Because $\|a + b\|^2 = \|a\|^2 + 2a^\top b + \|b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$

$$\mathbb{E}\|a + b\|^2 + \mathbb{E}\|a\|^2 \leq 3\mathbb{E}\|a\|^2 + 2\mathbb{E}\|b\|^2$$

Moreover, $1 + \frac{h}{2\kappa} \leq e^{\frac{h}{2\kappa}}$

$$\begin{aligned}
&\left(1 + \frac{h}{2\kappa}\right)e^{-\frac{h}{\kappa}}\mathbb{E}\left[\|y_{N-1} - x_{N-1}\|^2 + \|y_{N-1} + w_{N-1} - x_{N-1} - v_{N-1}\|^2\right] \\
&+ \frac{2\kappa}{h}\left(\mathbb{E}\|\mathbb{E}_\alpha x_N - x_N^*\|^2 + \mathbb{E}\|\mathbb{E}_\alpha x_N + v_N - x_N^* - v_N^*\|^2\right) \\
&+ \left(\mathbb{E}\|x_N - x_N^*\|^2 + \mathbb{E}\|x_N + v_N - x_N^* - v_N^*\|^2\right) \\
\leq &e^{-\frac{h}{2\kappa}}\mathbb{E}\left[\|y_{N-1} - x_{N-1}\|^2 + \|y_{N-1} + w_{N-1} - x_{N-1} - v_{N-1}\|^2\right] \\
&+ \frac{2\kappa}{h}\left(2\mathbb{E}\|\mathbb{E}_\alpha v_N - v_N^*\|^2 + 3\mathbb{E}\|\mathbb{E}_\alpha x_N - x_N^*\|^2\right) + \left(2\mathbb{E}\|v_N - v_N^*\|^2 + 3\mathbb{E}\|x_N - x_N^*\|^2\right)
\end{aligned}$$

This part of the proof is equivalent to

$$E_N \leq e^{-\frac{h}{2\kappa}}E_{N-1} + \delta_N$$

where,

- $E_N := \mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right]$
- $\delta_N := \frac{2\kappa}{h}\left(\mathbb{E}\|v_N - v_N^*\|^2 + 3\mathbb{E}\|x_N - x_N^*\|^2 + \mathbb{E}\|\mathbb{E}_\alpha x_N - x_N^*\|^2 + \mathbb{E}\|\mathbb{E}_\alpha[x_N + v_N] - x_N^* - v_N^*\|^2\right)$

Take recursion of E_N,

$$\begin{aligned}
E_N &\leq e^{-\frac{h}{2\kappa}} E_{N-1} + \delta_N \\
&\leq e^{-\frac{h}{2\kappa}} \left(e^{-\frac{h}{2\kappa}} E_{N-2} + \delta_{N-1} \right) + \delta_N = e^{-\frac{2h}{2\kappa}} E_{N-2} + (\delta_N + \delta_{N-1}) \\
&\leq e^{-\frac{3h}{2\kappa}} E_{N-3} + (\delta_N + \delta_{N-1} + \delta_{N-2}) \\
&\dots \\
&\leq e^{-\frac{Nh}{2\kappa}} E_0 + \sum_{n=1}^{N} e^{-\frac{(N-n)h}{2\kappa}} \delta_n \\
&\leq e^{-\frac{Nh}{2\kappa}} E_0 + \sum_{n=1}^{N} \delta_n \\
&= e^{-\frac{Nh}{2\kappa}} \mathbb{E} \left[\|y_0 - x_0\|^2 + \|y_0 + w_0 - x_0 - v_0\|^2 \right] \\
&+ \sum_{n=1}^{N} \frac{2\kappa}{h} \left(2\mathbb{E} \left\|\mathbb{E}_\alpha v_n - v_n^*\right\|^2 + 3\mathbb{E} \left\|\mathbb{E}_\alpha x_n - x_n^*\right\|^2 \right) \\
&+ \sum_{n=1}^{N} \left(2\mathbb{E} \left\|v_n - v_n^*\right\|^2 + 3\mathbb{E} \left\|x_n - x_n^*\right\|^2 \right)
\end{aligned}$$

B.3 Bound of $E_0 = \mathbb{E}\left[\|y_0 - x_0\|^2 + \|y_0 + w_0 - x_0 - v_0\|^2\right]$

By **Proposition 1** (C.6) of [4], $\mathbb{E}\|y_0 - x_0\|^2 \leq \frac{d}{m}$

Using the inequality $\|a + b\|^2 = \|a\|^2 + 2a^\top b + \|b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we obtain:

$$\mathbb{E}\|a + b\|^2 + \mathbb{E}\|a\|^2 \leq 2\mathbb{E}\|a\|^2 + 2\mathbb{E}\|b\|^2 + \mathbb{E}\|a\|^2 = 3\mathbb{E}\|a\|^2 + 2\mathbb{E}\|b\|^2.$$

$$\mathbb{E}\left[\|y_0 - x_0\|^2 + \|y_0 + w_0 - x_0 - v_0\|^2\right] \leq 3\mathbb{E}\|y_0 - x_0\|^2 + 2\mathbb{E}\|w_0 - v_0\|^2 \leq \frac{5d}{m}.$$

Plug in $N = \frac{2\kappa}{h} \log\left(\frac{20}{\epsilon^2}\right)$, which is our limiting iteration times,

$$e^{-\frac{Nh}{2\kappa}} \mathbb{E}\left[\|y_0 - x_0\|^2 + \|y_0 + w_0 - x_0 - v_0\|^2\right] \leq \exp\big(-\frac{2\kappa}{h} \log\left(\frac{20}{\epsilon^2}\right) \frac{h}{2\kappa}\big) \frac{5d}{m} = \frac{\epsilon^2 d}{4m}$$

B.4 Bound of δ_n

We first expand the term:

$$\sum_{n=1}^{N} \frac{2\kappa}{h} \left(2\mathbb{E}\|\mathbb{E}_\alpha v_n - v_n^*\|^2 + 3\mathbb{E}\|\mathbb{E}_\alpha x_n - x_n^*\|^2\right).$$

Now substitute the bounds:

$$\begin{aligned}
\mathbb{E}\|\mathbb{E}_\alpha v_n - v_n^*\|^2 &\leq O\left(h^8\|v_n\|^2 + u^2h^{10}\|\nabla f(x_n)\|^2 + udh^9\right), \\
\mathbb{E}\|\mathbb{E}_\alpha x_n - x_n^*\|^2 &\leq O\left(h^{10}\|v_n\|^2 + u^2h^{12}\|\nabla f(x_n)\|^2 + udh^{11}\right).
\end{aligned}$$

Then plug these into the sum:

$$\begin{aligned}
&\sum_{n=1}^{N} \frac{2\kappa}{h} \left(2\mathbb{E}\|\mathbb{E}_\alpha v_n - v_n^*\|^2 + 3\mathbb{E}\|\mathbb{E}_\alpha x_n - x_n^*\|^2\right) \\
&\leq \sum_{n=1}^{N} \frac{2\kappa}{h} \Big(2 \cdot O(h^8\|v_n\|^2 + u^2h^{10}\|\nabla f(x_n)\|^2 + udh^9) \\
&\quad + 3 \cdot O(h^{10}\|v_n\|^2 + u^2h^{12}\|\nabla f(x_n)\|^2 + udh^{11})\Big).
\end{aligned}$$

Now collect terms:

$$= O\left(\frac{2\kappa}{h} \sum_{n=1}^{N} \left((2h^8 + 3h^{10})\|v_n\|^2 + (2u^2h^{10} + 3u^2h^{12})\|\nabla f(x_n)\|^2 + (2udh^9 + 3udh^{11})\right)\right).$$

Now simplify the leading orders:

$$= O\left(h^7\kappa \sum_{n=1}^{N} \|v_n\|^2 + \frac{u}{m}h^9 \sum_{n=1}^{N} \|\nabla f(x_n)\|^2 + \frac{1}{m}Ndh^8\right).$$

We next expand the second term:

$$\sum_{n=1}^{N} \left(2\mathbb{E}\|v_n - v_n^*\|^2 + 3\mathbb{E}\|x_n - x_n^*\|^2\right).$$

Now substitute the bounds:

$$\begin{aligned}
\mathbb{E}\|v_n - v_n^*\|^2 &\leq O\left(h^4\|v_n\|^2 + u^2h^4\|\nabla f(x_n)\|^2 + udh^5\right), \\
\mathbb{E}\|x_n - x_n^*\|^2 &\leq O\left(h^6\|v_n\|^2 + u^2h^4\|\nabla f(x_n)\|^2 + udh^7\right).
\end{aligned}$$

Now plug these into the sum:

$$\begin{aligned}
&\sum_{n=1}^{N} \left(2\mathbb{E}\|v_n - v_n^*\|^2 + 3\mathbb{E}\|x_n - x_n^*\|^2\right) \\
&\leq \sum_{n=1}^{N} \Big(2 \cdot O(h^4\|v_n\|^2 + u^2h^4\|\nabla f(x_n)\|^2 + udh^5) \\
&\quad + 3 \cdot O(h^6\|v_n\|^2 + u^2h^4\|\nabla f(x_n)\|^2 + udh^7)\Big).
\end{aligned}$$

Now collect terms:

$$= O\left(\sum_{n=1}^{N} \left((2h^4 + 3h^6)\|v_n\|^2 + (2u^2h^4 + 3u^2h^4)\|\nabla f(x_n)\|^2 + (2udh^5 + 3udh^7)\right)\right).$$

Now simplify the leading orders:

$$= O\left(h^4 \sum_{n=1}^{N} \|v_n\|^2 + u^2h^4 \sum_{n=1}^{N} \|\nabla f(x_n)\|^2 + Nudh^5\right).$$

B.5 Bound of $\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2$ and $\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2$

First we need to calculate $\left|\mathbb{E}\nabla f(x_N)^\top v_N\right|$ in **Lemma 12** (C.4)

$$
\begin{aligned}
\left|\mathbb{E}\nabla f(x_N)^\top v_N\right| &\leq \mathbb{E}\left[\left|\nabla f(x_N)^\top v_N\right|\right] \\
&\leq \mathbb{E}\left[\|\nabla f(x_N)\| \cdot \|v_N\|\right] \quad \text{by Cauchy-Schwarz inequality } \left|a^\top b\right| \leq \|a\| \cdot \|b\| \\
&\leq \mathbb{E}\left[L\|v_N\|^2 + u\|\nabla f(x_N)\|^2\right] \quad \text{by Young's inequality } \|a\| \cdot \|b\| \leq L\|b\|^2 + \frac{1}{L}\|a\|^2
\end{aligned}
$$

Because

$$
\begin{aligned}
\|v_N\|^2 &= \|w_N + (v_N - w_N)\|^2 \leq 2\|w_N\|^2 + 2\|v_N - w_N\|^2 \\
\|\nabla f(x_N)\|^2 &= \|\nabla f(y_N) + (\nabla f(x_N) - \nabla f(y_N))\|^2 \\
&\leq 2\|\nabla f(y_N)\|^2 + 2\|\nabla f(x_N) - \nabla f(y_N)\|^2 \\
&\leq 2\|\nabla f(y_N)\|^2 + 2L^2\|x_N - y_N\|^2 \quad \text{by Lipschitz } \|\nabla f(x_N) - \nabla f(y_N)\| \leq L\|x_N - y_N\|
\end{aligned}
$$

and by Appendix A.3 and **Lemma 2** (C.5) of [2], we have $\mathbb{E}\|w_N\|^2 = \frac{d}{L}$ and $\mathbb{E}\|\nabla f(y_N)\|^2 \leq dL$,

$$
\begin{aligned}
\left|\mathbb{E}\nabla f(x_N)^\top v_N\right| &\leq 2\mathbb{E}\left[L\|w_N\|^2 + L\|v_N - w_N\|^2 + u\|\nabla f(y_N)\|^2 + L\|x_N - y_N\|^2\right] \\
&\leq 4d + 2L\mathbb{E}\left[\|v_N - w_N\|^2 + \|x_N - y_N\|^2\right]
\end{aligned}
$$

Since

$$
\begin{aligned}
\|v_N - w_N\|^2 &= \|(x_N + v_N) - (y_N + w_N) - (x_N - y_N)\|^2 \\
&\leq 2\|(x_N + v_N) - (y_N + w_N)\|^2 + 2\|x_N - y_N\|^2 \\
&\leq 3\|(x_N + v_N) - (y_N + w_N)\|^2 + 2\|x_N - y_N\|^2,
\end{aligned}
$$

we can obtain:

$$
\left|\mathbb{E}\nabla f(x_N)^\top v_N\right| \leq 4d + 6L\mathbb{E}\left[\|x_N - y_N\|^2 + \|(x_N + v_N) - (y_N + w_N)\|^2\right] = 4d + 6LE_N
$$

By **Lemma 12** (C.4),

$$
\begin{aligned}
\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 &\leq O\left(NLd + \frac{L}{h}\left|\mathbb{E}[\nabla f(x_N)^\top v_N]\right|\right) \\
\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 &\leq O\left(Nud + u\left|\mathbb{E}[\nabla f(x_N)^\top v_N]\right|\right)
\end{aligned}
$$

Then plug in $N = \frac{2\kappa}{h}\log(\frac{20}{\epsilon^2})$ and the upper bound of $\left|\mathbb{E}\nabla f(x_N)^\top v_N\right|$:

$$
\begin{aligned}
\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 &\leq O\left(\left(\frac{2\kappa}{h}\log\left(\frac{20}{\epsilon^2}\right)\right)Ld + \frac{L}{h}(4d + 6LE_N)\right), \\
&= O\left(\frac{\kappa dL}{h}\log\left(\frac{1}{\epsilon^2}\right) + \frac{L^2}{h}E_N\right), \\
&= O\left(\frac{\kappa dL}{h}\log\left(\frac{1}{\epsilon^2}\right) + \frac{L^2}{h}E_N\right).
\end{aligned}
$$

Similarly:

$$\begin{aligned}\sum_{n=0}^{N-1}\mathbb{E}\|v_n\|^2 &\leq O\left(\left(\frac{2\kappa}{h}\log\left(\frac{20}{\epsilon^2}\right)\right)ud + u(4d+6LE_N)\right),\\ &= O\left(\frac{\frac{L}{m}du}{h}\log\left(\frac{1}{\epsilon^2}\right) + uLE_N\right),\\ &= O\left(\frac{d}{hm}\log\left(\frac{1}{\epsilon^2}\right) + E_N\right).\end{aligned}$$

B.6 The final bound of $\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n)\|^2$ for the specific choice of h

To obtain the inequality $O\left(\left(\frac{\kappa dh^6}{m} + \frac{dh^3}{m}\right)\log\left(\frac{1}{\epsilon^2}\right)\right) \leq \frac{\epsilon^2 d}{4m}$ we must show that, with

$$h \;=\; C\;\min\Bigl(\underbrace{\varepsilon^{1/3}\,\kappa^{-1/6}\;\log^{-1/6}\bigl(1/\varepsilon^2\bigr)}_{A}\;,\;\underbrace{\varepsilon^{2/3}\;\log^{-1/3}\bigl(1/\varepsilon^2\bigr)}_{B}\Bigr),$$

for a sufficiently small constant $C > 0$, the following holds:

$$\bigl(\tfrac{\kappa\,d\,h^6}{m} + \tfrac{d\,h^3}{m}\bigr)\;\log\bigl(1/\varepsilon^2\bigr) \;\leq\; \frac{\varepsilon^2\,d}{4\,m}.$$

Since $d/m > 0$ is a common factor, it suffices to show

$$\kappa\,h^6\;\log\bigl(1/\varepsilon^2\bigr) \;+\; h^3\;\log\bigl(1/\varepsilon^2\bigr) \;\leq\; \frac{\varepsilon^2}{4}. \tag{1}$$

Set $L := \log(1/\varepsilon^2)$. Then (1) becomes

$$\kappa\,h^6\,L \;+\; h^3\,L \;\leq\; \frac{\varepsilon^2}{4}.$$

Because

$$h \;=\; C\,\min\{A, B\} \quad\text{with}\quad A \;=\; \varepsilon^{1/3}\,\kappa^{-1/6}\,L^{-1/6}, \quad B \;=\; \varepsilon^{2/3}\,L^{-1/3},$$

we split into two cases:

Case 1: $h = C\,A = C\,\varepsilon^{1/3}\,\kappa^{-1/6}\,L^{-1/6}$.

- *Bound $\kappa\,h^6\,L$.* Since

  $$h^6 = C^6\,\varepsilon^2\,\kappa^{-1}\,L^{-1}, \quad\Longrightarrow\quad \kappa\,h^6\,L = \kappa\left(C^6\,\varepsilon^2\,\kappa^{-1}\,L^{-1}\right)L = C^6\,\varepsilon^2.$$

  If $C^6 \leq \frac{1}{8}$, then $\kappa\,h^6\,L \leq \frac{\varepsilon^2}{8}$.

- *Bound $h^3\,L$.* Here

  $$h^3 = C^3\,\varepsilon\,\kappa^{-1/2}\,L^{-1/2}, \quad\Longrightarrow\quad h^3\,L = C^3\,\varepsilon\,\kappa^{-1/2}\,L^{1/2}.$$

  Since $\kappa^{-1/2} \leq 1$ (because $\kappa = L/m \geq 1$, where the L here refers to the L-lipschitz), we get

  $$h^3\,L \leq C^3\,\varepsilon\,L^{1/2} \leq C^3\,\varepsilon\,\frac{1}{\varepsilon} = C^3.$$

  If $C^3 \leq \frac{1}{8}$, then $h^3\,L \leq \frac{1}{8} \leq \frac{\varepsilon^2}{8}$ (since $\varepsilon \in (0,1)$ implies $\varepsilon^2 \leq 1$).

Combining these two bounds gives

$$\kappa\, h^6\, L + h^3\, L \;\le\; \frac{\varepsilon^2}{8} + \frac{\varepsilon^2}{8} = \frac{\varepsilon^2}{4}.$$

Hence in Case 1 it suffices to choose

$$C^3 \le \frac{1}{8} \quad \text{and} \quad C^6 \le \frac{1}{8},$$

for example $C = 2^{-1/3}$.

Case 2: $h = C\, B = C\, \varepsilon^{2/3}\, L^{-1/3}$.

- *Bound* $\kappa\, h^6\, L$*.* Now

  $$h^6 = C^6\, \varepsilon^4\, L^{-2}, \quad \Longrightarrow \quad \kappa\, h^6\, L = \kappa\big(C^6\, \varepsilon^4\, L^{-2}\big)\, L = C^6\, \kappa\, \frac{\varepsilon^4}{L}.$$

  For sufficiently small $\varepsilon \in (0,1)$, one sees $\varepsilon^4/L \le \varepsilon^2$. Hence $\kappa\, h^6\, L \le C^6\, \kappa\, \varepsilon^2$. But in Case 2 we also have $\varepsilon^{2/3}\, L^{-1/3} \le \kappa^{-1/7}$ (see below), which implies $\kappa\, \varepsilon^2 \le 1$. Therefore $\kappa\, h^6\, L \le C^6$. If $C^6 \le \frac{1}{8}$, then $\kappa\, h^6\, L \le \frac{1}{8} \le \frac{\varepsilon^2}{8}$.

- *Bound* $h^3\, L$*.* Here

  $$h^3 = C^3\, \varepsilon^2\, L^{-1}, \quad \Longrightarrow \quad h^3\, L = C^3\, \varepsilon^2.$$

  If $C^3 \le \frac{1}{8}$, then $h^3\, L \le \frac{\varepsilon^2}{8}$.

Hence in Case 2, choosing $C^3 \le \frac{1}{8}$ and $C^6 \le \frac{1}{8}$ again gives

$$\kappa\, h^6\, L + h^3\, L \;\le\; \frac{\varepsilon^2}{4}.$$

Verifying $\varepsilon^{2/3}\, L^{-1/3} \le \kappa^{-1/7}$ in Case 2. We need

$$\varepsilon^{2/3}\, L^{-1/3} \;\le\; \kappa^{-1/7} \quad \Longleftrightarrow \quad \varepsilon^2\, L^{-1/1} \;\le\; \kappa^{-2/7}.$$

Since $L = \ln(1/\varepsilon^2)$ grows more slowly than any power of $1/\varepsilon$, for all sufficiently small ε one has $\varepsilon^2\, L^{-1} \le \kappa^{-2/7}$. In other words, as $\varepsilon \to 0$, $\varepsilon^2/L \to 0$, forcing $\varepsilon^{2/3} L^{-1/3} \le \kappa^{-1/7}$. A similar argument shows $\varepsilon^{1/3}\, \kappa^{-1/6}\, L^{-1/6} \le \kappa^{-1/7}$ in Case 1 for small ε.

Conclusion. Taking

$$C \;=\; 2^{-1/3}, \qquad h \;=\; C\, \min\!\Big(\varepsilon^{1/3}\, \kappa^{-1/6}\, L^{-1/6}\;,\; \varepsilon^{2/3}\, L^{-1/3}\Big), \quad L := \ln\!\big(1/\varepsilon^2\big),$$

ensures in both Case 1 and Case 2 that

$$\kappa\, h^6\, L + h^3\, L \;\le\; \frac{\varepsilon^2}{4},$$

i.e.

$$\big(\tfrac{\kappa\, d\, h^6}{m} + \tfrac{d\, h^3}{m}\big)\, \ln\!\big(1/\varepsilon^2\big) \;\le\; \frac{\varepsilon^2\, d}{4\, m}.$$

This completes the proof of the desired bound.

B.7 The final bound of $\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2$ for the specific choice of h

To obtain the inequality

$$O(\kappa h^7 + h^3)E_N \;\leq\; \frac{1}{2}E_N,$$

we first divide E_N on both sides and simplify the $O(\cdot)$:

$$\kappa h^7 + h^3 \;\leq\; \frac{1}{2}.$$

Here, we separate the left hand side as the first part and the second part :

$$\text{First part} = \kappa h^7 \leq \frac{1}{4}, \qquad \text{Second part} = h^3 \leq \frac{1}{4}.$$

First part: Since $\kappa > 0$, we have $h \leq (\frac{1}{4\kappa})^{1/7}$.
Second part: We have $h \leq (\frac{1}{4})^{1/3}$.
Combining the two parts, we have

$$h \leq \min\left\{(\frac{1}{4\kappa})^{1/7},\, (\frac{1}{4})^{1/3}\right\},$$

with a constant $C \geq 0$,

$$h \leq C\min\left\{\kappa^{-1/7},\, 1\right\} = O(\kappa^{-1/7}).$$

That is, with any $h \leq O(\kappa^{-1/7})$, we have

$$\kappa h^7 + h^3 \leq \frac{1}{2}.$$

Next, we will show that we have (like what we did in the first half of this appendix B.6, where $L = \log(1/\varepsilon^2)$)

$$A = \varepsilon^{1/3}\kappa^{-1/6}L^{-1/6} \leq \kappa^{-1/7}, \qquad B = \varepsilon^{2/3}L^{-1/3} \leq \kappa^{-1/7} :$$

We start from $A \;\leq\; \kappa^{-1/7}$. Since $\kappa = \frac{L}{m} \geq 1$, which is commonly accepted with L-Lipschitz and m-strongly convex, we divide $\kappa^{-1/6}$ on both sides,

$$\varepsilon^{1/3}L^{-1/6} \;\leq\; \kappa^{1/42}.$$

With $\varepsilon \in (0, 1)$, $\varepsilon^{1/3}L^{-1/6} \leq 1$. Then as long as we have the ε,

$$\varepsilon^{1/3}L^{-1/6} \leq 1 \leq \kappa^{1/42} \Rightarrow A \leq \kappa^{1/7}$$

holds.
For $B = \varepsilon^{2/3}L^{-1/3} \leq \kappa^{-1/7}$, we can multipy $L^{1/3}$ on both sides:

$$\varepsilon^{2/3} \leq \kappa^{-1/7}L^{1/3}$$

Similar to the case of A, for any sufficiently small ε, we have

$$\varepsilon^{2/3}\,L^{-1/3} \;\leq\; \kappa^{-1/7},$$

since the whole left hand side would converge to 0.
Hence $B \leq \kappa^{-1/7}$ is proved.
As $A, B \leq \kappa^{-1/7}$ has been shown, we conclude that

$$h = C\min\left(\varepsilon^{1/3}\kappa^{-1/6}\log^{-1/6}\left(1/\varepsilon^2\right),\; \varepsilon^{2/3}\log^{-1/3}\left(1/\varepsilon^2\right)\right) \leq \kappa^{-1/7}.$$

C Lemma/Proposition Appendix

This appendix section contains all lemmas referenced throughout the paper. The lemmas/propositions are taken directly from their source(s), and the intuition and importance of each lemma have been added by the authors of this paper. Proofs for core lemmas have also been included.

C.1 Lemma 6 of Shen and Lee 2019 [6]

Let $x(t)$ and $v(t)$ be the solution to the ULD on $t \in [0, h]$. Assume that $h \leq \frac{1}{20}$ and $u = \frac{1}{L}$. Then we have the following bounds:

$$\mathbb{E}\Big[\sup_{t\in[0,h]} \|v(t)\|^2\Big] \;\leq\; O\Big(\,\|v(0)\|^2 \;+\; u^2h^2\,\|\nabla f(x(0))\|^2 \;+\; u\,d\,h\Big),$$

$$\mathbb{E}\Big[\sup_{t\in[0,h]} \|\nabla f(x(t))\|^2\Big] \;\leq\; O\Big(\,\|\nabla f(x(0))\|^2 \;+\; L^2h^2\,\|v(0)\|^2 \;+\; L\,d\,h^3\Big),$$

$$\mathbb{E}\Big[\sup_{t\in[0,h]} \|\,x(0) - x(t)\|^2\Big] \;\leq\; O\Big(h^2\,\|v(0)\|^2 \;+\; u^2h^4\,\|\nabla f(x(0))\|^2 \;+\; u\,d\,h^3\Big),$$

and

$$-\,\mathbb{E}\Big[\inf_{t\in[0,h]} \|v(t)\|^2\Big] \;\leq\; -\,\tfrac{1}{3}\,\|v(0)\|^2 \;+\; O\Big(u^2h^2\,\|\nabla f(x(0))\|^2 \;+\; u\,d\,h\Big),$$

$$-\,\mathbb{E}\Big[\inf_{t\in[0,h]} \|\nabla f(x(t))\|^2\Big] \;\leq\; -\,\tfrac{1}{3}\,\|\nabla f(x(0))\|^2 \;+\; O\Big(h^2L^2\,\|v(0)\|^2 \;+\; L\,d\,h^3\Big).$$

C.1.1 Importance of Lemma 6 of Shen and Lee 2019 [6]

In this Lemma 6, we show the properties of ULD, including their upper bounds and lower bounds. In the later proofs of the main theorem or of the lemmas, these properties serve as fundamental results: every occurrence in which one appeals to the upper or lower bounds of the ULD relies on this lemma.

- A bound on $\mathbb{E}\big[\sup_{0\leq t\leq h} \|v(t)\|^2\big]$, which prevents the velocity from growing uncontrolled over a single step of length h.
- A bound on $\mathbb{E}\big[\sup_{0\leq t\leq h} \|\nabla f(x(t))\|^2\big]$, ensuring that the gradient along the continuous trajectory remains uniformly $O\big(\|\nabla f(x(0))\|^2 + \cdots\big)$.
- A bound on $\mathbb{E}\big[\sup_{0\leq t\leq h} \|x(t) - x(0)\|^2\big]$, which controls the deviation $x_n^*(t)$ from its initial point x_n.
- Contraction estimates guarantee that both $\|v(t)\|^2$ and $\|\nabla f(x(t))\|^2$ decrease by at least a factor 1/3 over each interval of length h.

C.2 Lemma 1 of Shen and Lee 2019 [6]

Let (x_0, v_0) and (y_0, w_0) be two arbitrary points in $\mathbb{R}^d \times \mathbb{R}^d$. Let (x_t, v_t) and (y_t, w_t) be the exact solutions of the ULD after time t. If (x_t, v_t) and (y_t, w_t) are coupled through a shared Brownian motion, then,

$$\mathbb{E}\left[\|x_t - y_t\|^2 + \|(x_t + v_t) - (y_t + w_t)\|^2\right] \leq e^{-t/\kappa}\mathbb{E}\left[\|x_0 - y_0\|^2 + \|(x_0 + v_0) - (y_0 + w_0)\|^2\right].$$

Proof. [1]. Assume we have (x_t, v_t), (y_t, w_t) as two different initializations under a same ULD. $z_t = x_t - y_t$ is the difference between two position trajectories while $\psi_t = v_t - w_t$ is the difference between two velocity trajectories. Lyapunov function is defined as:

$$\mathcal{L}(t) = \mathbb{E}\left[\|z_t\|^2 + \|z_t + \psi_t\|^2\right],$$

Using the underdamped Langevin dynamics:

$$\begin{aligned} dz_t &= \psi_t \, dt \\ d\psi_t &= -\gamma \psi_t \, dt - \nabla f(x_t) + \nabla f(y_t) \end{aligned}$$

Compute $d\mathcal{L}(t)/dt$:

$$\begin{aligned} \frac{d}{dt}\mathcal{L}(t) = \frac{d}{dt}\mathbb{E}\left[\|z_t\|^2 + \|z_t + \psi_t\|^2\right] &= 2\,\mathbb{E}\left[\langle z_t, \psi_t\rangle + \langle z_t + \psi_t, \psi_t - \gamma\psi_t - (\nabla f(x_t) - \nabla f(y_t))\rangle\right] \\ &= 2\,\mathbb{E}\left[\langle z_t, \psi_t\rangle + \langle z_t + \psi_t, \psi_t\rangle - \gamma\langle z_t + \psi_t, \psi_t\rangle - \langle z_t + \psi_t, \nabla f(x_t) - \nabla f(y_t)\rangle\right] \\ &= 2\,\mathbb{E}\left[\langle z_t + z_t + \psi_t, \psi_t\rangle - \gamma\langle z_t + \psi_t, \psi_t\rangle - \langle z_t + \psi_t, \nabla f(x_t) - \nabla f(y_t)\rangle\right] \\ &= 2\,\mathbb{E}\left[(2-\gamma)\langle z_t + \psi_t, \psi_t\rangle - \langle z_t + \psi_t, \nabla f(x_t) - \nabla f(y_t)\rangle\right] \end{aligned}$$

Co-coercivity inequality describes a geometric property of the gradient of a smooth convex function. For an L-smooth convex function $f : \mathbb{R}^d \to \mathbb{R}$, the gradient satisfies the following inequality:

$$\langle x - y, \nabla f(x) - \nabla f(y)\rangle \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2$$

Now, use the ∇f is m-strongly convex and L-smooth with co-coercivity inequality:

$$\langle a, \nabla f(x) - \nabla f(y)\rangle \geq \frac{mL}{m+L}\|a\|^2 + \frac{1}{m+L}\|\nabla f(x) - \nabla f(y)\|^2$$

Let $\kappa = L/m$, and choose $\gamma = 2$ (which is used in the main theorem 3). Then the inequality can be simplified to:

$$\frac{d}{dt}\mathcal{L}(t) \leq -\frac{1}{\kappa}\mathcal{L}(t)$$

Apply Grönwall's inequality:

$$\frac{d\mathcal{L}}{\mathcal{L}} = -\frac{1}{\kappa}dt \quad \Rightarrow \quad \ln\mathcal{L}(t) = -\frac{t}{\kappa} + C \quad \Rightarrow \mathcal{L}(t) = \mathcal{L}(0)e^{-t/\kappa}$$

This proves:

$$\mathbb{E}[\|x_t - y_t\|^2 + \|(x_t + v_t) - (y_t + w_t)\|^2] \leq e^{-t/\kappa}\mathbb{E}[\|x_0 - y_0\|^2 + \|(x_0 + v_0) - (y_0 + w_0)\|^2]$$

□

C.2.1 Importance of Lemma 1 of Shen and Lee 2019 [6]

When comparing results from the algorithm to the exact dynamics in the Wasserstein distance, the analysis fundamentally relies on the convergence trend of the continuous process. This convergence behavior is formalized in Lemma 1.

C.3 Lemma 2 of Shen and Lee 2019 [6]

For each iteration n of Algorithm 1, let $\mathbb{E}_\alpha$ be the expectation taken over the random choice of α in iteration n. Let $\mathbb{E}$ be the expectation taken over other randomness in iteration n. Let $(x_n^*(t), v_n^*(t))_{t\in[0,h]}$ be the solution of the exact underdamped Langevin diffusion starting from (x_n, v_n) coupled through a shared Brownian motion with $x_{n+\frac{1}{2}}$, v_n, and x_{n+1}. Assume that $h \leq \frac{1}{20}$ and $u = \frac{1}{L}$. Then, x_{n+1} and v_{n+1} of Algorithm 1 satisfy

$$\mathbb{E}\left\|\mathbb{E}_\alpha x_{n+1} - x_n^*(h)\right\|^2 \leq \mathcal{O}\left(h^{10}\|v_n\|^2 + u^2h^{12}\|\nabla f(x_n)\|^2 + udh^{11}\right),$$

$$\mathbb{E}\left\|x_{n+1} - x_n^*(h)\right\|^2 \leq \mathcal{O}\left(h^{6}\|v_n\|^2 + u^2h^{4}\|\nabla f(x_n)\|^2 + udh^{7}\right),$$

$$\mathbb{E}\left\|\mathbb{E}_\alpha v_{n+1} - v_n^*(h)\right\|^2 \leq \mathcal{O}\left(h^{8}\|v_n\|^2 + u^2h^{10}\|\nabla f(x_n)\|^2 + udh^{9}\right),$$

$$\mathbb{E}\left\|v_{n+1} - v_n^*(h)\right\|^2 \leq \mathcal{O}\left(h^{4}\|v_n\|^2 + u^2h^{4}\|\nabla f(x_n)\|^2 + udh^{5}\right).$$

C.3.1 Importance of Lemma 2 of Shen and Lee 2019 [6]

Lemma 2 provides precise, iteration-wise upper bounds on how well the randomized midpoint updates approximate the exact ULD flow over one step. It shows that both the biased (x_{n+1}, v_{n+1}) and unbiased $(\mathbb{E}_\alpha[x_{n+1}], \mathbb{E}_\alpha[v_{n+1}])$ estimates of the position and velocity deviate from the true continuous-time solution $(x_n^*(h), v_n^*(h))$ by quantities of order $\mathcal{O}$. These high-order error estimates are essential because:

1. **Controlling of the bias-free error terms in δ_n**:

   In the proof of the main theorem (3), we need to bound

   $$\|\mathbb{E}_\alpha[x_{n+1} - x_n^*(h)\|^2 \quad \text{and} \quad \|\mathbb{E}_\alpha[v_{n+1} - v_n^*(h)\|^2,$$

   which appead (multiplied by $\frac{2\kappa}{h}$) when applying Young's inequality. Lemma 2 shows these terms are $\mathcal{O}\left(h^{10}\|v_n\|^2 + u^2h^{12}\|\nabla f(x_n)\|^2 + udh^{11}\right)$ for position and $\mathcal{O}\left(h^{8}\|v_n\|^2 + u^2h^{10}\|\nabla f(x_n)\|^2 + udh^{9}\right)$ for velocity. Without this high-order control, the accumulation $\sum_{n=1}^N \delta_n$ could not be driven below $\mathcal{O}(\epsilon^2)$.

2. **Bounding the raw one-step discretization error**:

   Lemma 2 also quantifies

   $$\mathbb{E}\left\|x_{n+1} - x_n^*(h)\right\|^2 \quad \text{and} \quad \mathbb{E}\left\|v_{n+1} - v_n^*(h)\right\|^2$$

   by $\mathcal{O}\left(h^{6}\|v_n\|^2 + u^2h^{4}\|\nabla f(x_n)\|^2 + udh^{7}\right)$ and $\mathcal{O}\left(h^{4}\|v_n\|^2 + u^2h^{4}\|\nabla f(x_n)\|^2 + udh^{5}\right)$, respectively. These bounds feed directly into the terms in the proof:

   $$2\mathbb{E}\left\|v_{n+1} - v_n^*(h)\right\|^2 \quad \text{and} \quad 3\mathbb{E}\left\|x_{n+1} - x_n^*(h)\right\|^2$$

   in the decomposition of E_N. By ensuring each one-step error is at most $\mathcal{O}(h^4)$ (up to factors depending on $\|v_n\|$ and $\|\nabla f(x_n)\|$), Lemma 2 lets us show that with a suitable choice of h, the total discretization error remains $\mathcal{O}(\epsilon^2)$ after N steps.

   Essentially, this lemma guarantees the randomized midpoint updates stay sufficiently close to the true continuous ULD trajectory – both on average (unbiased estimate) and in mean square. We are controlling cumulative discretization bias and variance and assisting guarantee of convergence rate in the main theorem (3).

C.4 Lemma 12 of Shen and Lee 2019 [6]

Assume h is smaller than some given constant. For each iteration $n = 0, \dots, N-1$, let (v_n, x_n) be the starting point of Algorithm 1 in iteration n. Then, the x_n in iteration $n = 0, \dots, N-1$ satisfies

$$\sum_{n=0}^{N-1} \mathbb{E}\|\nabla f(x_n)\|^2 \leq \mathcal{O}\left(NLd + \frac{L}{h}\left|\mathbb{E}\nabla f(x_N)^T v_N\right|\right).$$

Furthermore, the v_n in iteration $n = 0, \dots, N-1$ satisfies

$$\sum_{n=0}^{N-1} \mathbb{E}\|v_n\|^2 \leq \mathcal{O}\left(Nud + u\left|\mathbb{E}\nabla f(x_N)^T v_N\right|\right).$$

Proof. For each iteration $n = 0, \dots, N-1$, let $\{v_n(t),\, x_n(t)\}_{t\in[0,h]}$ be the exact underdamped Langevin diffusion starting from (v_n, x_n) computed in Algorithm 1. By definition,

$$\begin{aligned}
\mathbb{E}\big[\langle \nabla f(x_n(t)),\, v_n(t)\rangle\big] &= \mathbb{E}\big[v_n(t)^\top \nabla^2 f(x_n(t))\, v_n(t) \;+\; \nabla f(x_n(t))^\top\, dv_n(t)\big] \\
&= \mathbb{E}\big[v_n(t)^\top \nabla^2 f(x_n(t))\, v_n(t) \;-\; 2\nabla f(x_n(t))^\top v_n(t) \;-\; u\|\nabla f(x_n(t))\|^2\big].
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{E}\big[\nabla f(x_n(h))^\top v_n(h)\big] &= \mathbb{E}\Big[\nabla f(x_n(0))^\top v_n(0) \;+\; \int_0^h d\big(\nabla f(x_n(t))^\top v_n(t)\big)\Big] \\
&= \mathbb{E}\Big[\nabla f(x_n(0))^\top v_n(0) + \int_0^h \Big(v_n(t)^\top \nabla^2 f(x_n(t))\, v_n(t) \\
&- 2\nabla f(x_n(t))^\top v_n(t) - u\|\nabla f(x_n(t))\|^2\Big)\, dt\Big] \\
&\leq \mathbb{E}\Big[\nabla f(x_n(0))^\top v_n(0) + 3L\int_0^h \|v_n(t)\|^2 dt - \tfrac{1}{2}\, u \int_0^h \|\nabla f(x_n(t))\|^2 dt\Big] \\
&\leq \mathbb{E}\Big[\nabla f(x_n(0))^\top v_n(0) + 3Lh \sup_{t\in[0,h]} \|v_n(t)\|^2 - \tfrac{1}{2}\, hu \inf_{t\in[0,h]} \|\nabla f(x_n(t))\|^2\Big] \\
&\leq \mathbb{E}[\nabla f(x_n(0))^\top v_n(0)] \;-\; \tfrac{1}{6}\, hu\, \mathbb{E}\|\nabla f(x_n(0))\|^2 \\
&\quad + 3Lh \cdot O\big(\mathbb{E}\|v_n(0)\|^2 + u^2h^2\, \mathbb{E}\|\nabla f(x_n(0))\|^2 + udh\big) \\
&\leq \mathbb{E}[\nabla f(x_n(0))^\top v_n(0)] \;-\; \tfrac{1}{6}\, hu\, \mathbb{E}\|\nabla f(x_n(0))\|^2 \\
&\quad + O\big(Lh\, \mathbb{E}\|v_n(0)\|^2 + uh^2\, \mathbb{E}\|\nabla f(x_n(0))\|^2 + dh^2\big),
\end{aligned}$$

where the third step follows by Young's inequality, the fifth step by Lemma 6, and the last step since h is small. Also, we have

$$\begin{aligned}
&\mathbb{E}\Big[\nabla f(x_{n+1}(0))^\top v_{n+1}(0) - \nabla f(x_n(h))^\top v_n(h)\Big]\\
&= \mathbb{E}\big[(\nabla f(x_{n+1}(0)) - \nabla f(x_n(h)))^\top (v_{n+1}(0) - v_n(h))\big]\\
&\quad + \mathbb{E}\big[(\nabla f(x_{n+1}(0)) - \nabla f(x_n(h)))^\top v_n(h)\big]\\
&\le u\,\mathbb{E}\|\nabla f(x_{n+1}(0)) - \nabla f(x_n(h))\|^2 + Lh\,\mathbb{E}\|v_{n+1}(0) - v_n(h)\|^2 + uh^2\,\mathbb{E}\|\nabla f(x_n(h))\|^2\\
&\quad + \tfrac{L}{h^2}\,\mathbb{E}\|v_{n+1}(0) - v_n(h)\|^2 + \tfrac{u}{h}\,\mathbb{E}\|\nabla f(x_{n+1}(0)) - \nabla f(x_n(h))\|^2 + hL\,\mathbb{E}\|v_n(h)\|^2\\
&\le \tfrac{2u}{h}\,\mathbb{E}\|\nabla f(x_{n+1}(0)) - \nabla f(x_n(h))\|^2 + \tfrac{2L}{h^2}\,\mathbb{E}\|v_{n+1}(0) - v_n(h)\|^2 + uh^2\,\mathbb{E}\|\nabla f(x_n(h))\|^2 + hL\,\mathbb{E}\|v_n(h)\|^2\\
&\le \frac{2L}{h}\cdot O\big(h^6\mathbb{E}\|v_n(0)\|^2 + h^4u^2\mathbb{E}\|\nabla f(x_n(0))\|^2 + udh^7\big)\\
&\quad + \frac{2L}{h^2}\cdot O\big(h^4\mathbb{E}\|v_n(0)\|^2 + u^2h^2\mathbb{E}\|\nabla f(x_n(0))\|^2 + udh^5\big)\\
&\quad + uh^2\cdot O\big(\mathbb{E}\|\nabla f(x_n(0))\|^2 + L^2h^2\mathbb{E}\|v_n(0)\|^2 + Ldh^3\big)\\
&\quad + hL\cdot O\big(\mathbb{E}\|v_n(0)\|^2 + u^2h^2\mathbb{E}\|\nabla f(x_n(0))\|^2 + udh\big)\\
&\le O\big(hL\,\mathbb{E}\|v_n(0)\|^2 + uh^2\,\mathbb{E}\|\nabla f(x_n(0))\|^2 + dh^2\big),
\end{aligned}$$

where the second step uses Young's inequality and the fourth step uses Lemmas 2 and 6. Combining the above equations

$$\begin{aligned}
&\mathbb{E}[\nabla f(x_{n+1}(0))^\top v_{n+1}(0)]\\
&\le\ \mathbb{E}[\nabla f(x_n(0))^\top v_n(0)]\ -\ \tfrac{1}{6}\,hu\,\mathbb{E}\|\nabla f(x_n(0))\|^2\ +\ O\big(Lh\,\mathbb{E}\|v_n(0)\|^2 + uh^2\,\mathbb{E}\|\nabla f(x_n(0))\|^2 + dh^2\big).
\end{aligned}$$

Summing over $n = 0$ to $N-1$,

$$\begin{aligned}
&\sum_{n=0}^{N-1}\mathbb{E}[\nabla f(x_{n+1}(0))^\top v_{n+1}(0)]\\
&\le\ \sum_{n=0}^{N-1}\mathbb{E}[\nabla f(x_n(0))^\top v_n(0)] - \tfrac{1}{6}\,hu\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n(0))\|^2 + O\Big(Ndh + Lh\sum_{n=0}^{N-1}\mathbb{E}\|v_n(0)\|^2 + uh^2\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n(0))\|^2\Big).
\end{aligned}$$

Since $v_0 = 0$, one deduces

$$\tfrac{1}{8}\,hu\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n(0))\|^2\ \le\ O\big(Ndh + \big|\mathbb{E}[\nabla f(x_N(0))^\top v_N(0)]\big|\big),$$

which implies

$$\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n(0))\|^2\ \le\ O\Big(NLd + \tfrac{L}{h}\big|\mathbb{E}[\nabla f(x_N(0))^\top v_N(0)]\big|\Big).$$

By Lemma 11,

$$\sum_{n=0}^{N-1}\mathbb{E}\|v_n(0)\|^2\ \le\ O\Big(u^2h\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n(0))\|^2 + Nud\Big)\ \le\ O\big(Nud + u\big|\mathbb{E}[\nabla f(x_N(0))^\top v_N(0)]\big|\big).$$

□

C.4.1 Importance of Lemma 12 of Shen and Lee 2019 [6]

Lemma 12 provides a unified upper bound on the sum of squared gradient norms and the sum of squared velocities over all iterations. In the proof of the main theorem 3, we must control the global accumulation of discretization error.Lemma 12 shows that $\sum_n \|\nabla f(x_n)\|^2$ and $\sum_n \|v_n\|^2$ can each be bounded by $O\big(N\,L\,d\big)$ or $O\Big(\frac{L}{h}\,\big|\mathbb{E}[\nabla f(x_N)^\top v_N]\big|\Big)$.Consequently, Lemma 12 is the key result that guarantees—under a suitable choice of step size h and number of iterations N—that the algorithm converges to an ε-accurate approximation of the target distribution at a rate of$O\big(\kappa^{7/6}\,\varepsilon^{-1/3} + \kappa\,\varepsilon^{-2/3}\big)$.

C.5 Lemma 2 of Dalalyan 2017 [2]

If the function f is continuously differentiable and the gradient of f is Lipschitz with constant M, then

$$\int_{\mathbb{R}^p} \|\nabla f(x)\|_2^2\,\pi(x)\,dx \le Mp.$$

Proof. [2] To simplify notations, we prove the lemma for $p = 1$. The function $x \mapsto f'(x)$ being Lipschitz continuous is almost surely differentiable. Furthermore, it is clear that $|f''(x)| \le M$ for every x for which $f''(x)$ exists.

Take the derivative of $\pi(x) = \frac{1}{Z}e^{-f(x)}$,

$$\pi'(x) = -\frac{1}{Z}f'(x)e^{-f(x)} = f'(x)\pi(x)$$

Therefore, we using the identity $f'(x)\,\pi(x) = -\,\pi'(x)$, we get

$$\begin{aligned}\int_{\mathbb{R}} f'(x)^2\,\pi(x)\,\mathrm{d}x &= f'(0)\int_{\mathbb{R}} f'(x)\,\pi(x)\,\mathrm{d}x \;+\; \int_{\mathbb{R}}\Big(\int_0^x f''(y)\,\mathrm{d}y\Big)\,f'(x)\,\pi(x)\,\mathrm{d}x\\ &= -\,f'(0)\int_{\mathbb{R}} \pi'(x)\,\mathrm{d}x \;-\; \int_{\mathbb{R}}\Big(\int_0^x f''(y)\,\mathrm{d}y\Big)\,\pi'(x)\,\mathrm{d}x\\ &= -\int_0^\infty\int_0^x f''(y)\,\pi'(x)\,\mathrm{d}y\,\mathrm{d}x \;+\; \int_{-\infty}^0\int_0^x f''(y)\,\pi'(x)\,\mathrm{d}y\,\mathrm{d}x.\end{aligned}$$

In view of Fubini's theorem, we may swap the order of integration in each double integral. Noting that $\pi'(x) = \frac{\mathrm{d}}{\mathrm{d}x}\big(\pi(x)\big)$, we obtain

$$\int_{\mathbb{R}} f'(x)^2\,\pi(x)\,\mathrm{d}x = \int_0^\infty f''(y)\,\pi(y)\,\mathrm{d}y \;+\; \int_{-\infty}^0 f''(y)\,\pi(y)\,\mathrm{d}y \;\le\; M,$$

since $\big|f''(y)\big| \le M$ and $\int_0^\infty \pi(y)\,\mathrm{d}y + \int_{-\infty}^0 \pi(y)\,\mathrm{d}y = 1.$ □

C.5.1 Importance of Lemma 2 of Dalalyan 2017 [2]

This lemma provides an upper bound for $\mathbb{E}\,\|\nabla f(y_N)\|^2 \;\le\; d\,L$ and $\sum_{n=0}^{N-1}\mathbb{E}\|\nabla f(x_n)\|^2 \;\le\; NLd$. In the proof of the main theorem 3, Lemma 2 guarantees that—once the continuous ULD process has sufficiently mixed to approach the stationary distribution π—the second moment of the squared gradient norm is bounded by the constant Md. This, in turn, ensures that all subsequent discretization error terms involving $\mathbb{E}\|\nabla f(\cdot)\|^2$ are controlled.

C.6 Proposition 1 of Durmus and Moulines 2018 [4]

Assume H1(3) and H2(3).

(i) *For all $t \geq 0$ and $x \in \mathbb{R}^d$,*

$$\int_{\mathbb{R}^d} \|y - x^\star\|^2 \, P_t(x, dy) \leq \|x - x^\star\|^2 e^{-2mt} + (d/m)(1 - e^{-2mt}).$$

(ii) *The stationary distribution π satisfies $\int_{\mathbb{R}^d} \|x - x^\star\|^2 \pi(dx) \leq d/m$.*

(iii) *For any $x, y \in \mathbb{R}^d$ and $t > 0$, $W_2(\delta_x P_t, \delta_y P_t) \leq e^{-mt}\|x - y\|$.*

(iv) *For any $x \in \mathbb{R}^d$ and $t > 0$,*

$$W_2(\delta_x P_t, \pi) \leq e^{-mt} \left\{ \|x - x^\star\| + (d/m)^{1/2} \right\}.$$

C.6.1 Importance of Proposition 1 of Durmus and Moulines 2018 [4]

At a high level, Proposition 1 tells us two things about the continuous-time ULD:

1. **Exponential pull/contraction toward the center:** If you start two copies of the continuous ULD (let's call them trajectories A and B) from different points in space, then as time goes on, their positions (and velocities) get exponentially closer together. In other words, no matter where you begin, the diffusion has a built-in "contracting" behavior that drags you toward the same stationary law.

2. **Stationary law is sharply concentrated**: Once the continuous dynamics have run for a long time and reached their equilibrium ("stationary distribution"), the random particle sits within roughly a radius of $\sqrt{d/m}$ around the global minimizer $x^\star$. Equivalently, if you sample directly from the ULD's stationary distribution, you know $\mathbb{E}\big[\|y - x^\star\|^2\big] \leq \frac{d}{m}$.

Both points are crucial for our proof:

1. When we compare our discrete algorithm (randomized midpoint updates) to the ideal continuous process, we want to argue: "Even if the discretization makes a small error at each step, the continuous dynamics would have exponentially shrunk that error away if we had run them exactly." Because continuous ULD contracts exponentially, any mismatch introduced by discretization does not blow up; instead, it keeps getting "pulled back" toward the center. Without that contraction, a tiny drift per step could accumulate into a huge error.

2. At the very first step of our discrete algorithm, we initialize $x_0 = x^\star$ and $v_0 = 0$, whereas the continuous process (y_0, w_0) is drawn exactly from the ULD's stationary law. Proposition 1(ii) guarantees that $\mathbb{E}\,\|y_0 - x^\star\|^2 \leq \frac{d}{m}$. In other words, even though we started the discrete chain at the exact minimizer and the continuous chain at a random stationarity draw, they are "not far apart" in mean squared distance—specifically within order $\frac{d}{m}$. That is the quantitative bound we need for E_0. If instead the stationary distribution could be arbitrarily spread out, we would have no way to bound E_0, and our induction on "contract+discretization error" would fail from the start.

References

[1] Xi Cheng, Niladri Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped langevin mcmc: A non-asymptotic analysis. In *Conference on Learning Theory*, pages 300–323, 2018.

[2] Arnak S. Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent, 2017.

[3] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B*, 79(3):651–676, 2017.

[4] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm, 2018.

[5] Yuchen Ma, Yu Chen, Nicolas Flammarion, and Michael I Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, 2019.

[6] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling, 2019.

[7] Santosh S Vempala and Andre Wibisono. Collocation methods for optimization and sampling. *arXiv preprint arXiv:1909.05503*, 2019.