Fast parallel sampling under isoperimetry

Nima Anari ANARI@STANFORD.EDU
Stanford University
Sinho Chewi SCHEWI@IAS.EDU
Institute for Advanced Study
Thuy-Duong Vuong TDVUONG@STANFORD.EDU
Stanford University

Editors: Shipra Agrawal and Aaron Roth

Abstract

We show how to sample in parallel from a distribution π over $\mathbb{R}^d$ that satisfies a log-Sobolev inequality and has a smooth log-density, by parallelizing the Langevin (resp. underdamped Langevin) algorithms. We show that our algorithm outputs samples from a distribution $\hat{\pi}$ that is close to π in Kullback–Leibler (KL) divergence (resp. total variation (TV) distance), while using only $\log(d)^{O(1)}$ parallel rounds and $\widetilde{O}(d)$ (resp. $\widetilde{O}(\sqrt{d})$) gradient evaluations in total. This constitutes the first parallel sampling algorithms with TV distance guarantees.

For our main application, we show how to combine the TV distance guarantees of our algorithms with prior works and obtain RNC sampling-to-counting reductions for families of discrete distribution on the hypercube $\{\pm 1\}^n$ that are closed under exponential tilts and have bounded covariance. Consequently, we obtain an RNC sampler for directed Eulerian tours and asymmetric determinantal point processes, resolving open questions raised in prior works.

Keywords: Langevin Monte Carlo, isoperimetry, parallel computation, sampling, underdamped Langevin Monte Carlo

1. Introduction

In this paper, we study the problem of designing fast parallel algorithms for sampling from continuous distributions $\pi(x) \propto \exp(-V(x))$ over $x \in \mathbb{R}^d$. Designing efficient sampling algorithms is a ubiquitous problem, but the focus of most prior works has been to minimize sequential efficiency criteria, such as the total number of arithmetic operations or total queries to V and its derivatives (see Chewi (2024) for an exposition). In contrast, in this work we focus on parallel efficiency; roughly speaking, this means that we would like to have algorithms that sequentially take polynomial time, but can be run on a pool of polynomially many processors (e.g., as in the PRAM model of computation) in much less time, ideally polylogarithmic.

Our main result is to propose simple parallelizations of Langevin Monte Carlo (LMC) and underdamped Langevin Monte Carlo (ULMC), two of the most widely studied sequential sampling algorithms, and to prove that they run in $\log(d)^{O(1)}$ parallel iterations, under standard tractability criteria on π: that it satisfies a log-Sobolev inequality (LSI), and that its potential V is smooth, i.e., has Lipschitz gradients.

Theorem 1 (Informal main theorem) *Suppose that $\pi = \exp(-V)$ is a density on $\mathbb{R}^d$ that satisfies a log-Sobolev inequality and has a smooth potential V. Assume that we are given (approximate) oracle access to ∇V. Then, we can produce samples from a distribution $\hat{\pi}$ with the following guarantees.*

- *For LMC, $\hat{\pi}$ is close to π in Kullback–Leibler divergence, and the algorithm uses $\log^2(d)$ parallel iterations and $\widetilde{O}(d)$ processors and gradient evaluations.*
- *For ULMC, $\hat{\pi}$ is close to π in total variation divergence, and the algorithm uses $\log^2(d)$ parallel iterations and $\widetilde{O}(\sqrt{d})$ processors and gradient evaluations.*

For formal statements, see Theorem 13 and Theorem 20. Throughout this paper, when we refer to the number of iterations, we refer to the model of *adaptive complexity*: here, in each round, the algorithm makes a batch of queries to a first-order oracle for π (i.e., given a set of finite points $\mathcal{X} \subseteq \mathbb{R}^d$, the oracle outputs $(V(x), \nabla V(x))$ for each $x \in \mathcal{X}$), and the adaptive complexity measures the number of rounds. The gradient complexity measures the total number of points at which the first-order oracle is queried.

As an immediate corollary, we obtain parallel samplers for the class of well-conditioned log-concave distributions, i.e., those which satisfy

$$\beta I \succeq \nabla^2 V \succeq \alpha I \,,$$

for some constants $\alpha, \beta > 0$, where β is the smoothness parameter, and α is the parameter of strong log-concavity. This is because the LSI, a form of isoperimetric inequality, holds for all strongly log-concave distributions, due to the Bakry–Émery criterion (Bakry and Émery, 2006). However, the LSI is a weaker condition than strong log-concavity, and it applies to even many non-log-concave distributions such as Gaussian convolutions of distributions with bounded support (Bardet et al., 2018; Chen et al., 2021). In addition, unlike log-concavity, LSI is preserved under bounded perturbations and Lipschitz transformations of the log-density function.

The state-of-the-art prior to our work was a fast parallel algorithm due to Shen and Lee (2019), which produced Wasserstein-approximate samples from well-conditioned log-concave distributions. We improve on the state-of-the-art in three ways:

- We replace the strong log-concavity assumption with the weaker assumption that π satisfies a log-Sobolev inequality.
- We bound the error in KL divergence and TV distance, as opposed to the weaker notion of Wasserstein error. This difference is crucial for our main application, as explained in Section 1.3.
- Our results hold given only approximate access to ∇V, as opposed to exact access. This is again crucial in some of our applications as explained in Section 1.3.

Concurrent work. The concurrent work of Yu and Dalalyan (2024) studies parallelized randomized midpoint discretizations of LMC and ULMC, and we refer to the detailed discussion therein for a comparison with the present work. Here, we simply note that their work focuses on Wasserstein guarantees, whereas we focus on TV guarantees as needed for the applications in Section 1.3.

1.1. Algorithm

For the sake of exposition, here we describe the parallel LMC algorithm and defer the discussion of parallel ULMC to Section 3.2.1.

Our algorithm is based on a parallelized discretization of the Langevin diffusion. The continuous-time Langevin diffusion is the solution to the stochastic differential equation

$$dX_t = -\nabla V(X_t)\, dt + \sqrt{2}\, dB_t \tag{1}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion in $\mathbb{R}^d$. Langevin Monte Carlo (LMC) is a discretization of the continuous Langevin diffusion, defined by the following iteration:

$$X_{(n+1)h} - X_{nh} = -h\, \nabla V(X_{nh}) + \sqrt{2}\, (B_{(n+1)h} - B_{nh})\,, \tag{2}$$

where $h > 0$ is a parameter defining the step size.

If π satisfies a log-Sobolev inequality (LSI), then the law of the continuous-time Langevin diffusion converges to the target distribution π at time $t \approx \operatorname{poly}\log(d)$. The discretization error, measured for example in the total variation distance, between the continuous Langevin diffusion and the discrete process, scales like $\approx dh$, so the step size h is set to $1/d$, causing LMC to take $\widetilde{O}(d)$ iterations to converge. Our algorithm, explained in Algorithm 1, uses parallelization to speed up LMC, so that the step size is $\Omega(1)$ and the parallel depth is of the same order as the convergence time of the continuous Langevin diffusion, that is, of order $\operatorname{poly}\log(d)$.

The input to the algorithm is a (potentially random) starting point X_0, together with an "approximate score oracle" s, which is a function $\mathbb{R}^d \to \mathbb{R}^d$ that we can query, and which is assumed to be uniformly close to the gradient ∇V.

The main idea behind the algorithm is to turn the task of finding solutions to our (stochastic) differential equation into the task of finding fixed points of what is known as the Picard iteration. At a high level, Picard iteration takes a trajectory $(X_t)_{t\geq 0}$ and maps it to another trajectory $(X'_t)_{t\geq 0}$ given by

$$X'_t = X_0 - \int_0^t \nabla V(X_u)\, du + \sqrt{2}\, B_t\,.$$

Now if $X = X'$, then X is a solution to the Langevin diffusion. Thus, one might hope that starting from some trajectory X_0, and applying Picard iterations multiple times, the whole trajectory converges to the fixed point. The main benefit of Picard iteration is that ∇V or s can be queried at all points in parallel.

Note that the Picard iteration can be analogously defined for discrete-time dynamics such as LMC. Our main result shows that Picard iteration applied to the discretized Langevin diffusion (LMC) converges fast (in $\operatorname{poly}\log(d)$ Picard iterations) for trajectories defined over intervals of length at most h, where now h can be taken to be macroscopically large ($h = \Omega(1)$). We repeat this process until time $\operatorname{poly}\log(d)$, which requires $N = \operatorname{poly}\log(d)/h$ sequential iterations.

1.2. Analysis techniques

Many algorithms for solving stochastic differential equations, such as the Langevin dynamics $(X^*_t)_{t\geq 0}$, turn the problem into numerical integration. The main idea is to approximate the difference between $X^*_{(n+1)h} - X^*_{nh}$ using the trapezoidal rule, i.e.,

$$\begin{aligned} X^*_{(n+1)h} - X^*_{nh} &= -\int_{nh}^{(n+1)h} \nabla V(X^*_s)\, ds + \sqrt{2}\, (B_{(n+1)h} - B_{nh}) \\ &\approx -\sum h\, \nabla V(X^*_{s_i}) + \sqrt{2}\, (B_{(n+1)h} - B_{nh})\,. \end{aligned}$$

Algorithm 1 Parallelized Langevin dynamics

Input: $X_0 \sim \mu_0$, approximate score function $s : \mathbb{R}^d \to \mathbb{R}^d$ ($s \approx \nabla V$)
for $n = 0, \dots, N-1$ **do**
  for $m = 0, \dots, M$ *in parallel* **do**
    $X^{(0)}_{nh+mh/M} \leftarrow X_{nh}$
    Sample Brownian motion $B_{nh+mh/M} \leftarrow B_{nh} + \mathcal{N}(0, (mh/M)\, I)$
  end
  for $k = 0, \dots, K-1$ **do**
    for $m = 0, \dots, M$ *in parallel* **do**
      $X^{(k+1)}_{nh+mh/M} \leftarrow X_{nh} - \frac{h}{M} \sum_{m'=0}^{m-1} s(X^{(k)}_{nh+m'h/M}) + \sqrt{2}\,(B_{nh+mh/M} - B_{nh})$
    end
  end
  $X_{(n+1)h} \leftarrow X^{(K)}_{nh+h}$
end

Since we cannot access the idealized process X^*, we instead start with a rough estimate $X^{(0)}$ and iteratively refine our estimation to obtain $X^{(1)}, \dots, X^{(K)}$ that are closer and closer to the ideal X^*. The refined estimations are obtained via another application of the trapezoidal rule, i.e., $X^{(k)}_{s_i}$ is computed using $\int_{s \le s_i} \nabla V(X^{(k-1)}_s)\, ds$. This framework can be easily parallelized: $\nabla V(X^{(k)}_{s_i})$ for different i's can be computed in parallel using one processor for each s_i.

In Shen and Lee (2019), the points s_i at which to evaluate $\nabla V(X_{s_i})$ are chosen randomly; hence, their framework is known as the randomized midpoint method. Unfortunately, there seem to be fundamental barriers to obtaining KL or TV accuracy guarantees for randomized midpoint algorithms. To illustrate, while accuracy in 2-Wasserstein distance can be achieved using $\widetilde{O}(d^{1/3})$ gradient evaluations using a randomized midpoint algorithm (Shen and Lee, 2019, Algorithm 1), accuracy in KL or TV distance using $o(d^{1/2})$ gradient evaluations is not known.

We deviate from the approach of Shen and Lee (2019) by keeping the s_i fixed. This greatly simplifies the algorithm and its analysis and allows us to show that parallelized LMC converges to π in KL divergence using the interpolation method (Vempala and Wibisono, 2019), at the cost of using $\widetilde{O}(d)$ gradient evaluations instead of $\widetilde{O}(\sqrt{d})$[1] as in Shen and Lee (2019, Algorithm 2). In Section 3.2, we then show how to obtain a sampler, based on ULMC, which enjoys the same parallel complexity but uses only $\widetilde{O}(\sqrt{d})$ gradient evaluations, matching the state-of-the-art in Shen and Lee (2019).

For simplicity of exposition, assume that in Algorithm 1, the score function s is exactly ∇V. We will show via induction that

$$\mathbb{E}[\|\nabla V(X^{(K)}_{s_i}) - \nabla V(X^{(K-1)}_{s_i})\|^2] \lesssim \exp(-3.5K)\,, \tag{3}$$

where K is the depth of refinement. In other words, the approximation error decays exponentially fast with the parallel depth.

1. While Shen and Lee (2019, Algorithm 1) needs only $\widetilde{O}(d^{1/3})$ gradient evaluations, its parallel round complexity is also $\widetilde{\Theta}(d^{1/3})$, which doesn't align with our goal of getting $\mathrm{poly}\log(d)$ parallel round complexity. On the other hand, Shen and Lee (2019, Algorithm 2) uses $\mathrm{poly}\log(d)$ parallel rounds but needs $\widetilde{\Theta}(\sqrt{d})$ gradient evaluations (see Shen and Lee, 2019, Theorem 4).

To obtain the KL divergence bound, note that

$$X^{(K)}_{nh+(m+1)h/M} - X^{(K)}_{nh+mh/M} = -\frac{h}{M}\nabla V(X^{(K-1)}_{nh+mh/M}) + \sqrt{2}\,(B_{nh+(m+1)h/M} - B_{nh+mh/M})$$

and $\nabla V(X^{(K-1)}_{nh+mh/M})$ only depends on X_{nh} and the Brownian motion B_t for $t \le mh/M$. Let X_t, $mh/M \le t - nh \le (m+1)h/M$, be the interpolation of $X^{(K)}_{nh+mh/M}$ and $X^{(K)}_{nh+(m+1)h/M}$, i.e.,

$$X_t - X^{(K)}_{nh+mh/M} = -(t - nh - mh/M)\,\nabla V(X^{(K-1)}_{nh+mh/M}) + \sqrt{2}\,(B_t - B_{nh+mh/M})\,.$$

Then by a similar argument as in Vempala and Wibisono (2019), if $\mu_t := \mathrm{law}(X^{(K)}_t)$ we obtain

$$\begin{aligned}
\partial_t\,\mathsf{KL}(\mu_t \parallel \pi) &\le -\frac{3\alpha}{2}\,\mathsf{KL}(\mu_t \parallel \pi) + \mathbb{E}[\|\nabla V(X^{(K)}_t) - \nabla V(X^{(K-1)}_{nh+mh/M})\|^2] \\
&\le -\frac{3\alpha}{2}\,\mathsf{KL}(\mu_t \parallel \pi) + 2\,\mathbb{E}[\|\nabla V(X^{(K)}_t) - \nabla V(X^{(K)}_{nh+mh/M})\|^2] \\
&\quad + 2\,\mathbb{E}[\|\nabla V(X^{(K)}_{nh+mh/M}) - \nabla V(X^{(K-1)}_{nh+mh/M})\|^2]\,,
\end{aligned}$$

where α is the log-Sobolev constant (introduced in Definition 10). We can directly bound the third term using eq. (3). The second term can be bounded via a standard discretization analysis, noting that the time interval is only of size h/M. It leads to the bound

$$\mathbb{E}[\|\nabla V(X^{(K)}_t) - \nabla V(X^{(K)}_{nh+mh/M})\|^2] \lesssim \frac{dh}{M}\,, \tag{4}$$

where M is the number of discretization points, i.e., the number of parallel score queries in each round. Thus, from eq. (3) and eq. (4), by setting $K = \widetilde{O}(1)$ and $M = \widetilde{O}(d)$, we can set the step size $h = \Omega(1)$ so that the parallelized Langevin algorithm takes $\widetilde{O}(1)$ steps to converge to the target distribution π.

To obtain the total gradient complexity, note that at each round k, each processor m makes a single gradient query for $\nabla V(X^{(k)}_{nh+mh/M})$. The queries are then stored in a shared memory, and the sums $\sum_{m'=0}^{m-1} \nabla V(X^{(k)}_{nh+m'h/M})$ for $m = 0, 1, \dots, M$ can be computed with an additional parallel depth of $O(\log M)$ (but this step does not require further gradient queries, and hence does not contribute to the adaptive complexity).

Remark 2 *One may wonder if our results apply to distributions satisfying a weaker functional inequality such as the Poincaré inequality, instead of the LSI. Unfortunately, this is not the case since our analysis relies on the fact that the continuous-time Langevin diffusion converges to the target distribution π in time $\mathrm{poly}\log(d)$, which holds under the LSI but not under the weaker Poincaré inequality (see Chewi et al., 2021, for details).*

The above strategy based on the interpolation method no longer works for ULMC, so here we instead use an approach based on Girsanov's theorem. See Section 3.2.2 for details.

1.3. Applications

The main application of our results is to obtain fast parallel algorithms for several discrete sampling problems by refining the framework obtained by Anari et al. (2023). Recently, Anari et al. (2023) showed a parallel reduction from sampling to counting for *discrete* distributions on the hypercube $\{\pm 1\}^n$, by combining a faithful discretization of stochastic localization and fast parallel sampling algorithms for continuous distributions. For a discrete distribution μ over $\{\pm 1\}^n$, their reduction involves $\log n$ iterations, each involving sampling from $\tau_w \mu * \mathcal{N}(0, cI)$ where $\tau_w \mu$ is the exponential tilt of μ by the vector $w \in \mathbb{R}^n$, defined as:

$$\tau_w \mu(x) \propto \exp(\langle w, x \rangle)\, \mu(x)\,.$$

Anari et al. (2023) showed that for some appropriately chosen parameter $c = O(1)$, $\tau_w \mu * \mathcal{N}(0, cI)$ is a continuous and well-conditioned log-concave distribution for a wide class of discrete distributions μ of interest, i.e., those that are fractionally log-concave (see Alimohammadi et al., 2021, for a survey on fractional log-concavity). In this way, they obtained a parallel reduction to the problem of sampling from continuous and well-conditioned log-concave distributions.

The key technical challenge in their work is to control the propagation of errors resulting from the continuous sampler. Samples in an iteration become part of the external field w at future steps. Assuming only the bound on W_2 guaranteed by Shen and Lee (2019), these errors can, in the worst case, be blown up by a factor of $\mathrm{poly}(n)$ in each iteration, resulting in a quasipolynomial blowup by the end. As a result, Anari et al. (2023) only manage to obtain $\log(n)^{O(1)}$ parallel time by using $n^{O(\log n)}$, that is *quasipolynomially many*, processors (also known as a QuasiRNC algorithm). For some specific distributions μ, specifically strongly Rayleigh distributions (Anari et al., 2023), they circumvent this shortcoming by establishing a property they call transport-stability for the distribution of interest, but several other notable distributions such as Eulerian tours and asymmetric determinantal point processes fall outside the reach of this trick. Here, by replacing the W_2 guarantee of Shen and Lee (2019) with a TV distance guarantee, we entirely remove the need for transport-stability, turning the previous QuasiRNC algorithms into RNC algorithms.

Hence, our result implies an RNC-time sampler for a fractionally log-concave distribution[2] μ given access to an oracle which, given input $w \in \mathbb{R}^n$, approximately computes the partition function of $\tau_w \mu$. This holds more generally for all μ whose tilts have constantly bounded covariance, i.e., $\mathrm{cov}\, \tau_y \mu \preceq O(1)\, I$, analogous to Anari et al. (2023).

The normalizing factor or partition function of $\tau_w \mu$ is $\sum_{x \in \{\pm\}^n} \exp(\langle w, x \rangle)\, \mu(x)$. Viewed as a function of w, the partition function is also known as the Laplace transform of μ. We denote the log of the partition function, a.k.a. the log-Laplace transform, by $\mathcal{L}_\mu w = \log \sum_{x \in \{\pm\}^n} \exp(\langle w, x \rangle)\, \mu(x)$. By an abuse of notation, we expand the definition of the Laplace transform to all vectors $w \in (\mathbb{R} \cup \{\pm\infty\})^n$ as follows. Let S be the set of coordinates i where $w_i \in \{\pm\infty\}$, then:

$$\mathcal{L}_\mu w = \log \sum_{x \in \{\pm\}^n,\ \mathrm{sign}\, x_S = \mathrm{sign}\, w_S} \exp(\langle w_{-S}, x_{-S} \rangle)\, \mu(x)\,.$$

2. A distribution $\mu : \{\pm 1\}^n \to \mathbb{R}_{\geq 0}$ is α-fractionally log-concave iff $\forall y \in \mathbb{R}^n_{\geq 0} : \mathrm{cov}\, \tau_y \mu \leq \frac{1}{2\alpha} (I + \mathrm{diag}\, \mathrm{mean}(\tau_y \mu))$. We note that since $\mathrm{mean}(\tau_y \mu) \in [-1, 1]^n$, this implies $\mathrm{cov}\, \tau_y \mu \leq \frac{1}{\alpha} I$. See Alimohammadi et al. (2021); Anari et al. (2023) for more background.

Definition 3 (Approximate oracle for the Laplace transform) *We say that the oracle $\mathcal{O}(\cdot)$ ε-approximately computes the log-Laplace transform at μ if on input w, $\mathcal{O}$ outputs $\exp(\hat{\mathcal{L}})$ s.t.*

$$|\hat{\mathcal{L}} - \mathcal{L}_\mu w| \le \varepsilon\,.$$

Theorem 4 *Suppose that a distribution μ on $\{\pm 1\}^n$ has $\operatorname{cov} \tau_w \mu \preceq O(1)\, I$ for all $w \in \mathbb{R}^n$, and we have an oracle for $O(\varepsilon/\sqrt{n})$-approximately computing the log-Laplace transform of μ. Then we can sample from a distribution ε-close in total variation distance to μ, in $\log(n/\varepsilon)^{O(1)}$ time using $(n/\varepsilon)^{O(1)}$ processors.*

Thus, we improve upon Anari et al. (2023)'s reduction from sampling to counting in two ways:

- We remove the assumption that the distribution needs to satisfy a transport inequality, which is only known to hold for strongly Rayleigh distributions and partition-constraint strongly Rayleigh distributions (Anari et al., 2023). Under the weaker assumption of fractional log-concavity or bounded covariance under tilts, Anari et al. (2023) were only able to show a QuasiRNC reduction from sampling to counting, i.e., their sampling algorithm uses $\approx n^{\log n}$ processors.
- We only require an approximate counting oracle (see Theorem 3) instead of the exact counting oracle required by Anari et al. (2023).

Theorem 4 implies the following corollary about asymmetric determinantal point processes (DPPs) and Eulerian tours. For completeness, we recall the definitions of asymmetric DPPs and (directed) Eulerian tours.

Determinantal point processes. A determinantal point process (DPP) is a probability distribution over subsets $S \subseteq [n]$. It is parameterized by a matrix $L \in \mathbb{R}^{n\times n}$ with

$$\mathbb{P}(S) \propto \det(L_{S,S})\,,$$

with $L_{S,S}$ being the principal submatrix whose columns and rows are indexed by S. We call L the ensemble matrix. Note that we need $\det(L_{S,S}) \ge 0$ for all S for this definition to work. This is satisfied by any symmetric PSD L, which yields the traditional (symmetric) DPPs (Kulesza and Taskar, 2012), and more generally for any L whose symmetrization is PSD, that is $L + L^\intercal \succeq 0$, which are called asymmetric DPPs (Gartrell et al., 2019). Given a cardinality $k \in \mathbb{N}$, the k-DPP parameterized by L is a distribution over subsets S of size k, defined by conditioning the samples from the DPP to have size k.

We can view DPPs and k-DPPs as distributions over $\{\pm 1\}^n$ by letting $\mu(x) \propto \det(L_{S,S})$ where $S = \{i \in [n] : x_i = 1\}$.

Eulerian tour. An *Eulerian tour* is a circuit in a finite graph that visits every edge exactly once (revisiting vertices is allowed). A directed graph (or digraph) has an Eulerian tour if and only if every vertex has equal in-degree and out-degree, and all of its vertices with non-zero degrees belong to a single strongly connected component. Such graphs are called *Eulerian digraphs*. Sampling from the uniform distribution over Eulerian tours in digraph can be reduced to sampling from asymmetric DPPs in RNC-time (see Anari et al., 2023, §6).

Lemma 5 (Alimohammadi et al. (2021); Anari et al. (2023)) *The asymmetric DPPs and asymmetric k-DPPs satisfy the preconditions of theorem 4.*

Corollary 6 *Suppose that μ is an asymmetric DPP on a ground set of size n or the distribution of uniformly random Eulerian tours in a digraph of size n. Then, we can sample from a distribution ε-close in total variation distance to μ in time $\log(n/\varepsilon)^{O(1)}$ using $(n/\varepsilon)^{O(1)}$ processors.*

Hence, we resolve Anari et al. (2021)'s question about designing an RNC sampler for directed Eulerian tours.

Note that for the distributions studied in Anari et al. (2023), counting can be done exactly via determinant computations, or in other words, there is exact access to the log-Laplace transform. But there are several non-exact approximate counting techniques in the literature that can be efficiently parallelized. A notable one is Barvinok's polynomial interpolation method (see, e.g., Barvinok and Barvinok, 2021). As an example of a distribution where Barvinok's method can be applied, consider a distribution μ on the hypercube $\{\pm 1\}^n$ defined by a polynomial Hamiltonian: $\mu(x) = \exp(p(x))$. Barvinok and Barvinok (2021) showed that for quadratic and cubic polynomials p, assuming the coefficients of degree 2 and 3 terms are not too large (see Barvinok and Barvinok (2021) for exact conditions), $\sum_{x\in\{\pm\}^n} \mu(x)$ can be approximately computed in quasipolynomial time. It can be observed that the approximation algorithm can be parallelized into a QuasiRNC one since it simply involves computing $n^{\log n}$ separate quantities. We note that because the condition on p does not involve the linear terms, we can also apply the same algorithm to $\tau_w\mu$, whose potential differs from μ only in the linear terms. In other words, Barvinok's method gives us the oracle in Theorem 3. In the same paper, Barvinok and Barvinok (2021) prove that the partition functions of these models are root-free in a sector, a condition known as sector-stability, which is known to imply fractional log-concavity (Alimohammadi et al., 2021). As a result, by plugging in Barvinok's approximate counting algorithm into our result, we obtain QuasiRNC sampling algorithms, which at least in the case of cubic p were not known before.

2. Preliminaries

We let $\log$ denote the natural logarithm. For $x \in \mathbb{R}^d$, $\|x\|$ denotes the usual Euclidean norm of x.

For two distributions ρ and π, we use $\mathsf{TV}(\rho, \pi)$ to denote their total variation distance defined as $\sup\{\rho(E) - \pi(E) \mid E \text{ is an event}\}$.

A stronger notion of distance is the Kullback–Leibler (KL) divergence.

Definition 7 (Kullback–Leibler divergence) *For two probability densities ρ, π we define*

$$\mathsf{KL}(\rho \parallel \pi) = \mathbb{E}_\rho \log(\rho/\pi)\,.$$

We have the following relation between the KL divergence and TV distance, known as the Pinsker inequality.

$$\mathsf{TV}(\rho, \pi) \le \sqrt{\frac{1}{2}\,\mathsf{KL}(\rho \parallel \pi)}\,.$$

2.1. Log-concave distributions

Consider a density function $\pi : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ where $\pi(x) = \exp(-V(x))$. We call V the potential function for π. Throughout the paper, we will assume that V is twice continuously differentiable for simplicity of exposition.

Definition 8 (Smoothness) *For $\beta > 0$, we say π is β-smooth if the gradients of the potential are β-Lipschitz, that is*

$$\|\nabla V(x) - \nabla V(y)\| \leq \beta \|x - y\|, \qquad \text{for all } x, y \in \mathbb{R}^d.$$

For twice differentiable V, this is equivalent to

$$-\beta I \preceq \nabla^2 V \preceq \beta I.$$

When V is convex, we call π a log-concave density. A strengthening of this condition is:

Definition 9 (Strong log-concavity) *For $\alpha > 0$, we say π is α-strongly log-concave if*

$$0 \prec \alpha I \preceq \nabla^2 V.$$

2.2. Log-Sobolev and transport-entropy inequalities

Definition 10 (Log-Sobolev inequality) *We say π satisfies a log-Sobolev inequality (LSI) with constant α if for all smooth $f : \mathbb{R}^d \to \mathbb{R}$,*

$$\operatorname{ent}_\pi f^2 := \mathbb{E}_\pi[f^2 \log(f^2 / \mathbb{E}_\pi(f^2))] \leq \frac{2}{\alpha} \mathbb{E}_\pi[\|\nabla f\|^2].$$

By the Bakry–Émery criterion (Bakry and Émery, 2006), if π is α-strongly log-concave then π satisfies LSI with constant α. The right-hand side of the above inequality can also be written as the relative Fisher information.

Definition 11 (Relative Fisher information) *The relative Fisher information of ρ w.r.t. π is*

$$\mathsf{FI}(\rho \parallel \pi) = \mathbb{E}_\rho[\|\nabla \log(\rho/\pi)\|^2]. \tag{5}$$

The LSI is equivalent to the following relation between KL divergence and Fisher information:

$$\mathsf{KL}(\rho \parallel \pi) \leq \frac{1}{2\alpha} \mathsf{FI}(\rho \parallel \pi) \qquad \text{for all probability measures } \rho.$$

Indeed, take $f = \sqrt{\rho/\pi}$ in the above definition of the LSI.

Definition 12 (Wasserstein distance) *We denote by W_2 the Wasserstein distance between ρ and π, which is defined as*

$$W_2^2(\rho, \pi) = \inf\left\{\mathbb{E}_{(X,Y)\sim\Pi}[\|X - Y\|^2] \mid \Pi \text{ is a coupling of } \rho, \pi\right\},$$

where the infimum is over coupling distributions Π of (X, Y) such that $X \sim \rho, Y \sim \pi$.

The log-Sobolev inequality implies the following transport-entropy inequality, known as Talagrand's T_2 inequality (Otto and Villani, 2000):

$$\frac{\alpha}{2} W_2^2(\rho, \pi) \leq \mathsf{KL}(\rho \parallel \pi). \tag{6}$$

3. Parallel sampling guarantees

In this section, we formally state our main parallel sampling guarantees.

3.1. LMC

We state the formal version of Theorem 1 for LMC as Theorem 13. Our assumption throughout is that the score function s is a pointwise accurate estimate of ∇V:

Assumption 1 *The score function $s : \mathbb{R}^d \to \mathbb{R}$ satisfies $\|s(x) - \nabla V(x)\| \le \delta$ for all $x \in \mathbb{R}^d$.*

Theorem 13 *Suppose that V is β-smooth and π satisfies a log-Sobolev inequality with constant α, and the score function s is δ-accurate. Let $\kappa := \beta/\alpha$. Suppose*

$$\begin{aligned} &\beta h \le 1/10\,, && \delta \le 2\sqrt{\alpha}\varepsilon\,, && M \ge 7\max\{\kappa d/\varepsilon^2, \kappa^2\}\,, \\ &K \ge 2 + \log M\,, && && Nh \ge \alpha^{-1}\log\frac{2\,\mathsf{KL}(\mu_0 \parallel \pi)}{\varepsilon^2}\,. \end{aligned} \tag{7}$$

Then, the output distribution μ_{Nh} of Algorithm 1 satisfies

$$\max\Big\{\frac{\sqrt{\alpha}}{2}\,W_2(\mu_{Nh}, \pi), \mathsf{TV}(\mu_{Nh}, \pi)\Big\} \le \sqrt{\frac{\mathsf{KL}(\mu_{Nh} \parallel \pi)}{2}} \le \varepsilon.$$

To make the guarantee more explicit, we can combine it with the following well-known initialization bound, see, e.g., Dwivedi et al. (2019, §3.2).

Corollary 14 *Suppose that $\pi = \exp(-V)$ with $0 \prec \alpha I \preceq \nabla^2 V \preceq \beta I$, and let $\kappa := \beta/\alpha$. Let $x^\star$ be the minimizer of V. Then, for $\mu_0 = \mathcal{N}(x^\star, \beta^{-1} I)$, it holds that $\mathsf{KL}(\mu_0 \parallel \pi) \le \frac{d}{2}\log\kappa$.*

Consequently, setting

$$h = \frac{1}{10\beta}\,, \quad \delta = 2\sqrt{\alpha}\varepsilon\,, \quad M = 7\max\Big\{\frac{\kappa d}{\varepsilon^2}, \kappa^2\Big\}\,, \quad K = 3\log M\,, \quad N = 10\kappa\log\frac{d\log\kappa}{\varepsilon^2}\,,$$

then Algorithm 1 initialized at μ_0 outputs μ_{Nh} satisfying

$$\max\Big\{\frac{\sqrt{\alpha}}{2}\,W_2(\mu_{Nh}, \pi), \mathsf{TV}(\mu_{Nh}, \pi)\Big\} \le \sqrt{\frac{\mathsf{KL}(\mu_{Nh} \parallel \pi)}{2}} \le \varepsilon\,.$$

Also, Algorithm 1 uses a total of $KN = \widetilde{O}(\kappa\log^2(d/\varepsilon^2))$ parallel rounds and M δ-approximate gradient evaluations in each round.

The proofs for this section are given in §A.

3.2. ULMC

In this section, we design a parallel sampler based on underdamped Langevin Monte Carlo (ULMC), also called *kinetic Langevin*, which has similar parallel iteration complexity as LMC but requires less total work. Since there are difficulties applying the interpolation method without higher-order smoothness assumptions (see the discussion in Ma et al. (2021); Zhang et al. (2023)), we will use a different proof technique based on Girsanov's theorem, as in Zhang et al. (2023); Altschuler and Chewi (2024). Note that since we seek TV guarantees, we cannot apply the coupling arguments of Cheng et al. (2018); Dalalyan and Riou-Durand (2020).

3.2.1. ALGORITHM

In continuous time, the underdamped Langevin diffusion is the coupled system of SDEs

$$\begin{aligned} dX_t &= P_t \, dt \,, \\ dP_t &= -\nabla V(X_t) \, dt - \gamma P_t \, dt + \sqrt{2\gamma} \, dB_t \,, \end{aligned}$$

where $\gamma > 0$ is the friction parameter. Throughout, we will simply set $\gamma = \sqrt{8\beta}$, where β is the smoothness parameter.

The idea for developing a parallel sampler is similar as before: we parallelize Picard iteration. However, in order to eventually apply Girsanov's theorem to analyze the algorithm, the discretization must be chosen so that $dX_t = P_t \, dt$ is preserved. Hence, we will use the exponential Euler integrator.

We use the following notation: $\tau(t)$ is the largest multiple of h/M which is less than t, i.e., $\tau(t) = \lfloor t / \frac{h}{M} \rfloor \frac{h}{M}$. We define a sequence of processes $(X^{(0)}, P^{(0)})$, $(X^{(1)}, P^{(1)})$, etc., so that

$$\begin{aligned} dX_t^{(k+1)} &= P_t^{(k+1)} \, dt \,, \\ dP_t^{(k+1)} &= -\nabla V(X_{\tau(t)}^{(k)}) \, dt - \gamma P_t^{(k+1)} \, dt + \sqrt{2\gamma} \, dB_t \,. \end{aligned}$$

This is a linear SDE, so it can be integrated exactly, yielding

$$\begin{aligned} X_{nh+(m+1)h/M}^{(k+1)} &= X_{nh+mh/M}^{(k+1)} + \frac{1 - \exp(-\gamma h/M)}{\gamma} \, P_{nh+mh/M}^{(k+1)} \\ &\quad - \frac{h/M - (1 - \exp(-\gamma h/M))/\gamma}{\gamma} \, \nabla V(X_{nh+mh/M}^{(k)}) + \xi_{n,m}^X \,, \end{aligned} \tag{8}$$

$$P_{nh+(m+1)h/M}^{(k+1)} = \exp(-\gamma h/M) \, P_{nh+mh/M}^{(k+1)} - \frac{1 - \exp(-\gamma h/M)}{\gamma} \, \nabla V(X_{nh+mh/M}^{(k)}) + \xi_{n,m}^P \,, \tag{9}$$

where $(\xi_{n,m}^X, \xi_{n,m}^P)$ is a correlated Gaussian vector in $\mathbb{R}^d \times \mathbb{R}^d$ with law $\mathcal{N}(0, \Sigma)$, where

$$\Sigma = \begin{bmatrix} \frac{2}{\gamma} \left[\frac{h}{M} - \frac{2}{\gamma} \left(1 - \exp(-\gamma h/M)\right) + \frac{1}{2\gamma} \left(1 - \exp(-2\gamma h/M)\right) \right] & * \\ \frac{1}{\gamma} \left(1 - 2\exp(-\gamma h/M) + \exp(-2\gamma h/M)\right) & 1 - \exp(-2\gamma h/M) \end{bmatrix} , \tag{10}$$

and the upper-left entry marked $*$ is determined by symmetry.

Note that each processor $m = 1, \ldots, M$ can independently generate a correlated Gaussian vector according to the above law and store it. Then, the updates for the above discretization can be computed quickly in parallel. We summarize the algorithm below as Algorithm 2.

3.2.2. ANALYSIS

We now give our guarantees for Algorithm 2. Compared to Theorem 13, it improves the number of processors by roughly a factor of $\sqrt{\kappa d}/\varepsilon$. Although it is stated for strongly log-concave measures for simplicity, similarly to §3.1, the discretization guarantees only require π to satisfy a log-Sobolev inequality and smoothness; see Theorem 20 for a more precise statement. The proof is given in §B.

Algorithm 2 Parallelized underdamped Langevin dynamics

Input: $(X_0, P_0) \sim \mu_0$, approximate score function $s : \mathbb{R}^d \to \mathbb{R}^d$ ($s \approx \nabla V$)
for $n = 0, \dots, N-1$ **do**
  for $m = 0, \dots, M$ *in parallel* **do**
    $(X^{(0)}_{nh+mh/M}, P^{(0)}_{nh+mh/M}) \leftarrow (X_{nh}, P_{nh})$
    Sample correlated Gaussian vectors according to eq. (10)
  end
  for $k = 0, \dots, K-1$ **do**
    for $m = 0, \dots, M$ *in parallel* **do**
      Compute $(X^{(k+1)}_{nh+mh/M}, P^{(k+1)}_{nh+mh/M})$ using eq. (8) and eq. (9), replacing ∇V with s
    end
  end
  $(X_{(n+1)h}, P_{(n+1)h}) \leftarrow (X^{(K)}_{nh+h}, P^{(K)}_{nh+h})$
end

Algorithm 3 Framework for discrete sampling via continuous sampling

Initialize $w_0 \leftarrow 0$
for $i = 0, \dots, T-1$ **do**
  $x_{i+1} \leftarrow$ (approximate) sample from $\tau_{w_i}\mu * \mathcal{N}(0, cI)$
  $w_{i+1} \leftarrow w_i + x_{i+1}/c$
end
return $\operatorname{sign} w_T \in \{\pm 1\}^n$

Theorem 15 *Assume that V is α-strongly convex and β-smooth; let $\kappa := \beta/\alpha$. Assume that V is minimized at $x^\star$. Consider Algorithm 2 initialized at $\mu_0 = \mathcal{N}(x^\star, \beta^{-1}I) \otimes \mathcal{N}(0, I)$ and with*

$$h = \Theta\big(1/\sqrt{\beta}\big)\,, \quad \delta \le \widetilde{O}\big(\frac{\sqrt{\alpha}\varepsilon}{\sqrt{\log d}}\big)\,, \quad M = \widetilde{\Theta}\big(\frac{\sqrt{\kappa d}}{\varepsilon}\big)\,, \quad K = \Theta\big(\log \frac{\kappa d}{\varepsilon^2}\big)\,, \quad N = \widetilde{\Theta}\big(\kappa \log \frac{d}{\varepsilon^2}\big)\,.$$

Then, the law of the output of Algorithm 2 is ε-close in total variation distance to π. The algorithm uses a total of $KN = \widetilde{\Theta}(\kappa \log^2(d/\varepsilon^2))$ parallel rounds and M δ-approximate gradient evaluations in each round.

4. Implications for sampling from discrete distributions

In this section, we prove Theorem 4. For simplicity, we only state our parallel guarantees using parallel LMC, for which the initialization is more straightforward, but it is easy to combine the results of this section with parallel ULMC as well. For concreteness, we restate Anari et al. (2023)'s sampling-to-counting reduction. Then, Theorem 4 is a consequence of Anari et al. (2023, Lemma 7), our fast parallel sampler with TV guarantee, and a modified version of Anari et al. (2023, Proposition 27). We include the proofs for completeness in §C.

We give the overall algorithm as Algorithm 3. The following lemma shows that the step of sampling from distributions of the form $\tau_w\mu * \mathcal{N}(0, cI)$ is a well-conditioned log-concave sampling problem, and moreover, that the score can be approximated quickly in parallel.

Lemma 16 (Anari et al. (2023)) *Let $\nu = \tau_w \mu * \mathcal{N}(0, cI)$. Then, $\nu \propto \exp(-V)$ with*

$$-\nabla V(y) = \frac{\operatorname{mean}(\tau_{y/c+w}\mu)}{c} - \frac{y}{c} = \frac{1}{c}\,\frac{\sum_{x\in\{\pm\}^n} x \exp(\langle y/c + w, x\rangle)\,\mu(x)}{\sum_{x\in\{\pm\}^n} \exp(\langle y/c + w, x\rangle)\,\mu(x)} - \frac{y}{c}$$

and

$$\nabla^2 V(y) = -\frac{\operatorname{cov} \tau_{y/c+w}\mu}{c^2} + \frac{I}{c}\,.$$

If $\operatorname{cov} \tau_y \mu \preceq \frac{c}{2} I$ for all $y \in \mathbb{R}^n$, then ν is well-conditioned strongly log-concave with condition number $\kappa = O(1)$, i.e., for all $y \in \mathbb{R}^n$:

$$\frac{1}{2c}\, I \preceq \nabla^2 V(y) \preceq \frac{1}{c}\, I\,.$$

Furthermore, a δ-approximate score function s for ∇V can be computed in $O(1)$ parallel iterations using n machines, each making $O(1)$ calls to an $\varepsilon = O(\delta\sqrt{c/n})$-approximate oracle for the Laplace transform of μ.

The next lemma states that if the samples from the continuous densities $\tau_w \mu * \mathcal{N}(0, cI)$ are accurate, then the output of Algorithm 3 outputs an approximate sample from μ.

Lemma 17 (Anari et al. (2023, Lemma 7)) *If the continuous samples are exact in Algorithm 3, then for $T = \Omega(c \log(n/\varepsilon))$, the distribution of cw_T/T is $\mu * \mathcal{N}(0, \frac{c}{T} I)$ and output of the algorithm is ε-close in total variation distance to μ.*

These results, together with an initialization bound (see Theorem 23), then yield the proof of Theorem 4. Details are given in §C.

Acknowledgments

SC acknowledges the support of the Eric and Wendy Schmidt Fund at the Institute for Advanced Study.

References

Yeganeh Alimohammadi, Nima Anari, Kirankumar Shiragur, and Thuy-Duong Vuong. Fractionally log-concave and sector-stable polynomials: counting planar matchings and more. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 433–446, New York, NY, USA, 2021. Association for Computing Machinery.

Jason M. Altschuler and Sinho Chewi. Faster high-accuracy log-concave sampling via algorithmic warm starts. *Journal of the ACM*, 3 2024.

Nima Anari, Nathan Hu, Amin Saberi, and Aaron Schild. Sampling arborescences in parallel. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 83:1–83:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

Nima Anari, Yizhi Huang, Tianyu Liu, Thuy-Duong Vuong, Brian Xu, and Katherine Yu. Parallel discrete sampling via continuous walks. In *STOC'23—Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 103–116. ACM, New York, 2023.

Dominique Bakry and Michel Émery. Diffusions hypercontractives. In *Séminaire de Probabilités XIX 1983/84: Proceedings*, pages 177–206. Springer, 2006.

Jean-Baptiste Bardet, Nathaël Gozlan, Florent Malrieu, and Pierre-André Zitt. Functional inequalities for Gaussian convolutions of compactly supported measures: explicit bounds and dimension dependence. *Bernoulli*, 24(1):333–353, 2018.

Alexander Barvinok and Nicholas Barvinok. More on zeros and approximation of the Ising partition function. In *Forum of Mathematics, Sigma*, volume 9, page e46. Cambridge University Press, 2021.

Hong-Bin Chen, Sinho Chewi, and Jonathan Niles-Weed. Dimension-free log-Sobolev inequalities for mixture distributions. *Journal of Functional Analysis*, 281(11):109236, 2021.

Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: a non-asymptotic analysis. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 300–323. PMLR, 7 2018.

Sinho Chewi. *Log-concave sampling*. Forthcoming, 2024. Available online at https://chewisinho.github.io/.

Sinho Chewi, Murat A. Erdogdu, Mufan (Bill) Li, Ruoqi Shen, and Matthew S. Zhang. Analysis of Langevin Monte Carlo from Poincaré to log-Sobolev. *arXiv preprint 2112.12662*, 2021.

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

Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu. Log-concave sampling: Metropolis–Hastings algorithms are fast. *Journal of Machine Learning Research*, 20(183):1–42, 2019.

Mike Gartrell, Victor-Emmanuel Brunel, Elvis Dohmatob, and Syrine Krichene. Learning nonsymmetric determinantal point processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Jean-François Le Gall. *Brownian motion, martingales, and stochastic calculus*, volume 274 of *Graduate Texts in Mathematics*. Springer, [Cham], French edition, 2016.

Yi-An Ma, Niladri S. Chatterji, Xiang Cheng, Nicolas Flammarion, Peter L. Bartlett, and Michael I. Jordan. Is there an analog of Nesterov acceleration for gradient-based MCMC? *Bernoulli*, 27(3): 1942 – 1992, 2021.

Pierre Monmarché. An entropic approach for Hamiltonian Monte Carlo: the idealized case. *The Annals of Applied Probability*, 34(2):2243 – 2293, 2024.

Felix Otto and Cédric Villani. Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality. *J. Funct. Anal.*, 173(2):361–400, 2000.

Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8094–8106. Curran Associates, Inc., 2019.

Cédric Villani. Hypocoercivity. *Mem. Amer. Math. Soc.*, 202(950):iv+141, 2009.

Lu Yu and Arnak S. Dalalyan. Parallelized midpoint randomization for Langevin Monte Carlo. *arXiv preprint 2402.14434*, 2024.

Matthew S. Zhang, Sinho Chewi, Mufan (Bill) Li, Krishnakumar Balasubramanian, and Murat A. Erdogdu. Improved discretization analysis for underdamped Langevin Monte Carlo. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 36–71. PMLR, 7 2023.

Appendix A. Proofs for LMC

In this section, we give the proofs for §3.1. Let $\mu_{nh} := \mathrm{law}(X_{nh})$. We first need the following recursive bound, which shows that the error decays exponentially fast in the parallel refinement.

Lemma 18 *Suppose that V is β-smooth, and that the score function s is δ-accurate. Assume that $\beta h \le 1/10$ and that π satisfies Talagrand's T_2 inequality with constant α. Then,*

$$\max_{m=1,\dots,M} \mathbb{E}[\|X^{(K)}_{nh+mh/M} - X^{(K-1)}_{nh+mh/M}\|^2] \le 34 \exp(-3.5K) \left(1.4dh + \frac{8\beta^2 h^2}{\alpha} \mathsf{KL}(\mu_{nh} \parallel \pi)\right) + 8.2\delta^2 h^2 \,.$$

Proof Let

$$\mathcal{E}_k := \max_{m=1,\dots,M} \mathbb{E}[\|X^{(k)}_{nh+mh/M} - X^{(k-1)}_{nh+mh/M}\|^2] \,.$$

For any $m = 1, \dots, M$,

$$\begin{aligned}
\mathbb{E}[\|X^{(k+1)}_{nh+mh/M} - X^{(k)}_{nh+mh/M}\|^2] &= \mathbb{E}\Big[\Big\| \frac{h}{M} \sum_{m'=1}^{m-1} \big(s(X^{(k)}_{nh+m'h/M}) - s(X^{(k-1)}_{nh+m'h/M})\big)\Big\|^2\Big] \\
&\le \frac{h^2 m}{M^2} \sum_{m'=1}^{m-1} \mathbb{E}[\|s(X^{(k)}_{nh+m'h/M}) - s(X^{(k-1)}_{nh+m'h/M})\|^2] \\
&\le 3h^2 \max_{m'=1,\dots,m} \mathbb{E}[\|\nabla V(X^{(k)}_{nh+m'h/M}) - \nabla V(X^{(k-1)}_{nh+m'h/M})\|^2] + 6\delta^2 h^2 \\
&\le 3\beta^2 h^2 \, \mathcal{E}_k + 6\delta^2 h^2
\end{aligned}$$

and hence $\mathcal{E}_{k+1} \le 3\beta^2 h^2 \, \mathcal{E}_k + 6\delta^2 h^2$. Also,

$$\begin{aligned}
\mathbb{E}[\|X^{(1)}_{nh+mh/M} - X_{nh}\|^2] &= \frac{h^2 m^2}{M^2} \mathbb{E}[\|s(X_{nh})\|^2] + \frac{dhm}{M} \\
&\le 2\delta^2 h^2 + 2h^2 \, \mathbb{E}[\|\nabla V(X_{nh})\|^2] + dh
\end{aligned}$$

and thus $\mathcal{E}_1$ is bounded by the right-hand side above. Iterating the recursion and using $\beta h \le 1/10$,

$$\begin{aligned}
\mathcal{E}_K &\le \exp(-3.5\,(K-1))\, \mathcal{E}_1 + 6.2\delta^2 h^2 \\
&\le \exp(-3.5\,(K-1))\, \{2\delta^2 h^2 + 2h^2 \, \mathbb{E}[\|\nabla V(X_{nh})\|^2] + dh\} + 6.2\delta^2 h^2 \,.
\end{aligned}$$

Also, by Vempala and Wibisono (2019, Lemma 10),

$$\mathbb{E}[\|\nabla V(X_{nh})\|^2] \le 2\beta d + \frac{4\beta^2}{\alpha} \mathsf{KL}(\mu_{nh} \parallel \pi) \,.$$

Substituting this in and using $\beta h \le 1/10$ yields the result. ∎

Proof [Proof of Theorem 13] We will use the interpolation method. Let $X_{nh+mh/M} = X^{(K)}_{nh+mh/M}$. It is easy to see that

$$X_{nh+(m+1)h/M} = X_{nh+mh/M} - \frac{h}{M} s(X^{(K-1)}_{nh+mh/M}) + \sqrt{2}\,(B_{nh+(m+1)h/M} - B_{nh+mh/M}) \,.$$

Let X denote the interpolation of $X^{(K)}$, i.e., for $t \in [nh + mh/M, nh + (m+1)h/M]$, let

$$X_t = X_{nh+mh/M} - (t - nh - mh/M)\, s(X^{(K-1)}_{nh+mh/M}) + \sqrt{2}\,(B_t - B_{nh+mh/M})\,.$$

Note that $s(X^{(K-1)}_{nh+mh/M})$ is a constant vector field given $X^{(K-1)}_{nh+mh/M}$. Let μ_t be the law of X_t. The same argument as in Vempala and Wibisono (2019, Proof of Lemma 3) yields the differential inequality

$$\begin{aligned}\partial_t \,\mathsf{KL}(\mu_t \parallel \pi) &= -\,\mathsf{FI}(\mu_t \parallel \pi) + \mathbb{E}\Big\langle \nabla V(X_t) - s(X^{(K-1)}_{nh+mh/M}), \nabla \log \frac{\mu_t(X_t)}{\pi(X_t)} \Big\rangle \\ &\le -\frac{3}{4}\,\mathsf{FI}(\mu_t \parallel \pi) + \mathbb{E}[\|\nabla V(X_t) - s(X^{(K-1)}_{nh+mh/M})\|^2]\end{aligned} \tag{11}$$

where we used $\langle a, b\rangle \le \|a\|^2 + \frac{1}{4}\,\|b\|^2$ and $\mathbb{E}[\|\nabla \log \frac{\mu_t(X_t)}{\pi(X_t)}\|^2] = \mathsf{FI}(\mu_t \parallel \pi)$. Next, we bound

$$\begin{aligned}&\mathbb{E}[\|\nabla V(X_t) - s(X^{(K-1)}_{nh+mh/M})\|^2] \\ &\quad\le 2\,\mathbb{E}[\|\nabla V(X_t) - \nabla V(X^{(K-1)}_{nh+mh/M})\|^2 + \|\nabla V(X^{(K-1)}_{nh+mh/M}) - s(X^{(K-1)}_{nh+mh/M})\|^2] \\ &\quad\le 2\,\mathbb{E}[\|\nabla V(X_t) - \nabla V(X^{(K-1)}_{nh+mh/M})\|^2] + 2\delta^2 \\ &\quad\le 2\beta^2\,\mathbb{E}[\|X_t - X^{(K-1)}_{nh+mh/M}\|^2] + 2\delta^2\,.\end{aligned} \tag{12}$$

Moreover,

$$\mathbb{E}[\|X_t - X^{(K-1)}_{nh+mh/M}\|^2] \le 2\,\mathbb{E}[\|X_t - X_{nh+mh/M}\|^2] + 2\,\mathbb{E}[\|X^{(K)}_{nh+mh/M} - X^{(K-1)}_{nh+mh/M}\|^2]\,. \tag{13}$$

The first term above is

$$\begin{aligned}\mathbb{E}[\|X_t - X_{nh+mh/M}\|^2] &= (t - nh - mh/M)^2\,\mathbb{E}[\|s(X^{(K-1)}_{nh+mh/M})\|^2] + d\,(t - nh - mh/M) \\ &\le \frac{2h^2}{M^2}\,\mathbb{E}[\|\nabla V(X^{(K-1)}_{nh+mh/M})\|^2] + \frac{2\delta^2 h^2}{M^2} + \frac{dh}{M} \\ &\le \frac{4\beta^2 h^2}{M^2}\,\mathbb{E}[\|X_t - X^{(K-1)}_{nh+mh/M}\|^2] + \frac{4h^2}{M^2}\,\mathbb{E}[\|\nabla V(X_t)\|^2] + \frac{2\delta^2 h^2}{M^2} + \frac{dh}{M}\,.\end{aligned}$$

Substituting this into eq. (13) and using $\beta h \le 1/10$ yields

$$\begin{aligned}\mathbb{E}[\|X_t - X^{(K-1)}_{nh+mh/M}\|^2] \le \frac{4.4h^2}{M^2}\,\mathbb{E}[\|\nabla V(X_t)\|^2] + \frac{2.2\delta^2 h^2}{M^2} + \frac{1.1dh}{M} \\ + 2.2\,\mathbb{E}[\|X^{(K)}_{nh+mh/M} - X^{(K-1)}_{nh+mh/M}\|^2]\,.\end{aligned}$$

Now, Chewi et al. (2021, Lemma 16) yields

$$\mathbb{E}[\|\nabla V(X_t)\|^2] \le \mathsf{FI}(\mu_t \parallel \pi) + 2\beta d\,.$$

For the last term, we can apply Theorem 18.

Substituting everything into eq. (11) and cleaning up the terms yields

$$\partial_t \mathsf{KL}(\mu_t \parallel \pi) \le -0.66\, \mathsf{FI}(\mu_t \parallel \pi) + 2.5\delta^2 + 2\beta^2 \left[\frac{2dh}{M} + 75\exp(-3.5K)\left(1.4dh + \frac{8\beta^2 h^2}{\alpha}\,\mathsf{KL}(\mu_{nh} \parallel \pi)\right)\right].$$

Assuming that $K \ge 1.3 + 0.3\log M$, and using the LSI,

$$\partial_t \mathsf{KL}(\mu_t \parallel \pi) \le -1.3\alpha\, \mathsf{KL}(\mu_t \parallel \pi) + 2.5\delta^2 + \frac{6.8\beta^2 dh}{M} + \frac{16\beta^4 h^2}{\alpha M}\,\mathsf{KL}(\mu_{nh} \parallel \pi).$$

Integrating this inequality,

$$\mathsf{KL}(\mu_{(n+1)h} \parallel \pi) \le \left[\exp(-1.3\alpha h) + \frac{16\beta^4 h^3}{\alpha M}\right] \mathsf{KL}(\mu_{nh} \parallel \pi) + 2.5\delta^2 h + \frac{6.8\beta^2 dh^2}{M}.$$

Provided $M \ge 6.4\kappa^2$, then $\exp(-1.3\alpha h) + \frac{16\beta^4}{h^3}\alpha M \le \exp(-\alpha h)$. Iterating,

$$\mathsf{KL}(\mu_{Nh} \parallel \pi) \le \exp(-\alpha N h)\, \mathsf{KL}(\mu_0 \parallel \pi) + \frac{2.8\delta^2}{\alpha} + \frac{7.5\beta^2 dh}{\alpha M}.$$

Thus we obtain the guarantee in KL divergence. The guarantees in TV and W_2 distance follow from Pinsker's and Talagrand's inequality respectively. ■

Appendix B. Proofs for ULMC

We turn towards the analysis of Algorithm 2. We start by bounding the discretization error between the algorithm and the continuous-time process using Girsanov's theorem. Throughout, let μ_{Nh} denote the law of the output of the algorithm, and let π_t denote the marginal law of the continuous-time Langevin diffusion at time t started from μ_0.

First, we need a lemma.

Lemma 19 *Let $(X_t, P_t)_{t\ge 0}$ denote the continuous-time underdamped Langevin diffusion, started at $(X_0, P_0) \sim \mu_0$. Assume that V is β-smooth, and that $\pi^X \propto \exp(-V)$ satisfies Talagrand's T_2 inequality with constant α. Let $\pi = \pi^X \otimes \mathcal{N}(0, I)$. Then,*

$$\mathbb{E}[\|\nabla V(X_t)\|^2] \le 2\beta d + \frac{4\beta^2}{\alpha}\,\mathsf{KL}(\mu_0 \parallel \pi), \qquad \mathbb{E}[\|P_t\|^2] \le 2d + \mathsf{KL}(\mu_0 \parallel \pi).$$

Proof For the first bound, we use a similar proof as Vempala and Wibisono (2019, Lemma 10). Namely, by Lipschitzness of ∇V, the transport inequality, and the data-processing inequality,

$$\begin{aligned}\mathbb{E}[\|\nabla V(X_t)\|^2] &\le 2\,\mathbb{E}_{\pi^X}[\|\nabla V\|^2] + 2\beta^2 W_2^2(\mathrm{law}(X_t), \pi^X) \le 2\beta d + \frac{4\beta^2}{\alpha}\,\mathsf{KL}(\mathrm{law}(X_t) \parallel \pi^X) \\ &\le 2\beta d + \frac{4\beta^2}{\alpha}\,\mathsf{KL}(\mathrm{law}(X_t, P_t) \parallel \pi) \le 2\beta d + \frac{4\beta^2}{\alpha}\,\mathsf{KL}(\mu_0 \parallel \pi).\end{aligned}$$

Similarly,

$$\mathbb{E}[\|P_t\|^2] \le 2\,\mathbb{E}_{\mathcal{N}(0,I)}[\|\cdot\|^2] + 2\,W_2^2(\mathrm{law}(P_t), \mathcal{N}(0,I)) \le 2d + 4\,\mathsf{KL}(\mu_0 \parallel \pi)\,.$$

This completes the proof. ∎

We now state and prove our main discretization bound.

Theorem 20 *Suppose that V is β-smooth and that $\pi^X \propto \exp(-V)$ satisfies Talagrand's T_2 inequality with constant α. Let $\kappa := \beta/\alpha$. Assume that the parallel depth satisfies $K \gtrsim \log M$ (for a sufficiently large implied constant) and that $h \lesssim 1/\sqrt{\beta}$ (for a sufficiently small implied constant). Then, it holds that*

$$\mathsf{KL}(\pi_T \parallel \mu_T) \lesssim \frac{T}{\sqrt{\beta}} \Big(\delta^2 + \frac{\beta^2 d h^2}{M^2} + \frac{\beta^2 h^2}{M^2}\big(1 + \frac{\kappa}{M^2}\big)\,\mathsf{KL}(\mu_0 \parallel \pi)\Big)\,.$$

Proof Let $\mathbf{P}$ denote the Wiener measure on $[0,T]$, under which $(B_t)_{t\in[0,T]}$ is a standard Brownian motion. Using this Brownian motion, we define the algorithm process, i.e.,

$$\begin{aligned} dX_t^{(k+1)} &= P_t^{(k+1)}\,dt\,,\\ dP_t^{(k+1)} &= -s(X_{\tau(t)}^{(k)})\,dt - \gamma P_t^{(k+1)}\,dt + \sqrt{2\gamma}\,dB_t\,. \end{aligned}$$

We also drop the superscripts for parallel depth K, i.e., $(X_t^{(K)}, P_t^{(K)}) = (X_t, P_t)$. We now write

$$dP_t = -\nabla V(X_t)\,dt - \gamma P_t\,dt + \sqrt{2\gamma}\,d\tilde{B}_t$$

where $d\tilde{B}_t = dB_t - \frac{1}{\sqrt{2\gamma}}\,(s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t))\,dt$. By Girsanov's theorem (see Le Gall, 2016, §5.6), if we define the path measure $\mathbf{Q}$ via

$$\frac{d\mathbf{Q}}{d\mathbf{P}} = \exp\Big(\frac{1}{\sqrt{2\gamma}}\int_0^T \langle s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t), dB_t\rangle - \frac{1}{8\gamma}\int_0^T \|s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2\,dt\Big)\,, \tag{14}$$

then under $\mathbf{Q}$ the process $\tilde{B}$ is a standard Brownian motion. It follows readily that under $\mathbf{Q}$, the process (X, P) is the continuous-time underdamped Langevin diffusion. By the data-processing inequality and eq. (14),

$$\begin{aligned} \mathsf{KL}(\pi_T \parallel \mu_T) &\le \mathsf{KL}(\mathbf{Q} \parallel \mathbf{P}) = \mathbb{E}_{\mathbf{Q}} \log \frac{d\mathbf{Q}}{d\mathbf{P}}\\ &= \mathbb{E}_{\mathbf{Q}}\Big[\frac{1}{\sqrt{2\gamma}}\int_0^T \langle s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t), dB_t\rangle - \frac{1}{8\gamma}\int_0^T \|s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2\,dt\Big]\\ &= \mathbb{E}_{\mathbf{Q}}\Big[\frac{1}{\sqrt{2\gamma}}\int_0^T \langle s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t), d\tilde{B}_t\rangle + \frac{1}{8\gamma}\int_0^T \|s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2\,dt\Big]\\ &= \frac{1}{8\gamma}\,\mathbb{E}_{\mathbf{Q}}\int_0^T \|s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2\,dt\,. \end{aligned} \tag{15}$$

From now on, all expectations are taken under $\mathbf{Q}$ and we drop the subscript $\mathbf{Q}$ from the notation. We focus on t lying in the interval $[nh, (n+1)h]$.

Of course, using the fact that we have δ-accurate gradient evaluations,

$$\begin{aligned}\mathbb{E}[\|s(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2] &\lesssim \delta^2 + \mathbb{E}[\|\nabla V(X_{\tau(t)}^{(K-1)}) - \nabla V(X_t)\|^2] \\ &\leq \delta^2 + \beta^2 \, \mathbb{E}[\|X_{\tau(t)}^{(K-1)} - X_t\|^2] \,. \end{aligned} \tag{16}$$

We split this into two terms:

$$\mathbb{E}[\|X_{\tau(t)}^{(K-1)} - X_t\|^2] \lesssim \mathbb{E}[\|X_t - X_{\tau(t)}\|^2] + \mathbb{E}[\|X_{\tau(t)} - X_{\tau(t)}^{(K-1)}\|^2] \,. \tag{17}$$

We begin with the recursive term (the second one).

For any $k = 1, \ldots, K$, let

$$\mathcal{E}_k \coloneqq \max_{m=1,\ldots,M} \mathbb{E}[\|X_{nh+mh/M}^{(k)} - X_{nh+mh/M}^{(k-1)}\|^2] \,.$$

To bound this quantity, we start with

$$\begin{aligned}\mathbb{E}[\|X_{nh+mh/M}^{(k)} - X_{nh+mh/M}^{(k-1)}\|^2] &= \mathbb{E}\Big[\Big\|\int_{nh}^{nh+mh/M} (P_t^{(k)} - P_t^{(k-1)}) \, dt\Big\|^2\Big] \\ &\leq h \int_{nh}^{nh+mh/M} \mathbb{E}[\|P_t^{(k)} - P_t^{(k-1)}\|^2] \, dt \,. \end{aligned} \tag{18}$$

Next,

$$\begin{aligned}\mathbb{E}[\|P_t^{(k)} - P_t^{(k-1)}\|^2] &= \mathbb{E}\Big[\Big\|\int_{nh}^{t} \{-(s(X_{\tau(s)}^{(k-1)}) - s(X_{\tau(s)}^{(k-2)})) - \gamma \, (P_s^{(k)} - P_s^{(k-1)})\} \, ds\Big\|^2\Big] \\ &\lesssim h \int_{nh}^{t} \mathbb{E}[\|s(X_{\tau(s)}^{(k-1)}) - s(X_{\tau(s)}^{(k-2)})\|^2 + \gamma^2 \, \|P_s^{(k)} - P_s^{(k-1)}\|^2] \, ds \,. \end{aligned}$$

By Grönwall's inequality,

$$\mathbb{E}[\|P_t^{(k)} - P_t^{(k-1)}\|^2] \lesssim h \exp(O(\gamma^2 h^2)) \int_{nh}^{t} \mathbb{E}[\|s(X_{\tau(s)}^{(k-1)}) - s(X_{\tau(s)}^{(k-2)})\|^2] \, ds \,.$$

Recall that $\gamma^2 \asymp \beta$. We assume throughout that $h \lesssim 1/\sqrt{\beta}$ for a sufficiently small implied constant, so that $\gamma^2 h^2 \lesssim 1$. Therefore,

$$\begin{aligned}\mathbb{E}[\|P_t^{(k)} - P_t^{(k-1)}\|^2] &\lesssim h \int_{nh}^{t} \mathbb{E}[\|s(X_{\tau(s)}^{(k-1)}) - s(X_{\tau(s)}^{(k-2)})\|^2] \, ds \\ &\lesssim \delta^2 h^2 + h \int_{nh}^{t} \mathbb{E}[\|\nabla V(X_{\tau(s)}^{(k-1)}) - \nabla V(X_{\tau(s)}^{(k-2)})\|^2] \, ds \\ &\lesssim \delta^2 h^2 + \beta^2 h \int_{nh}^{t} \mathbb{E}[\|X_{\tau(s)}^{(k-1)} - X_{\tau(s)}^{(k-2)}\|^2] \, ds \leq \delta^2 h^2 + \beta^2 h^2 \, \mathcal{E}_{k-1} \,. \end{aligned}$$

Substituting this into eq. (18), we obtain

$$\mathcal{E}_k \lesssim \delta^2 h^4 + \beta^2 h^4 \, \mathcal{E}_{k-1} \,.$$

Using $h \lesssim 1/\sqrt{\beta}$ and iterating this bound,

$$\mathcal{E}_K \lesssim \exp(-\Omega(K))\,\mathcal{E}_1 + \delta^2 h^4\,. \tag{19}$$

We must now bound $\mathcal{E}_1$. To do so, we note that

$$\mathbb{E}[\|X^{(1)}_{nh+mh/M} - X_{nh}\|^2] = \mathbb{E}\Big[\Big\|\int_{nh}^{nh+mh/M} P^{(1)}_t\,dt\Big\|^2\Big] \le h\int_{nh}^{nh+mh/M} \mathbb{E}[\|P^{(1)}_t\|^2]\,dt\,. \tag{20}$$

Also,

$$\begin{aligned}
\mathbb{E}[\|P^{(1)}_t\|^2] &\lesssim \mathbb{E}[\|P_{nh}\|^2] + \mathbb{E}\Big[\Big\|\int_{nh}^{t}\{-s(X_{nh}) - \gamma P^{(1)}_s\}\,ds + \sqrt{2\gamma}\,(B_t - B_{nh})\Big\|^2\Big] \\
&\lesssim \mathbb{E}[\|P_{nh}\|^2] + h^2\,\mathbb{E}[\|s(X_{kh})\|^2] + \gamma^2 h\int_{nh}^{t}\mathbb{E}[\|P^{(1)}_s\|^2]\,ds \\
&\quad + \gamma\,\mathbb{E}[\|\tilde{B}_t - \tilde{B}_{nh}\|^2] + \mathbb{E}\Big[\Big\|\int_{nh}^{t}(s(X^{(K-1)}_{\tau(s)}) - \nabla V(X_s))\,ds\Big\|^2\Big] \\
&\lesssim \mathcal{P} + h^2\delta^2 + h^2\mathcal{G} + \gamma^2 h\int_{nh}^{t}\mathbb{E}[\|P^{(1)}_s\|^2]\,ds + \gamma dh + h^2\Delta\,.
\end{aligned}$$

In the above bound, we were careful to recall that we are working under $\mathbf{Q}$, for which $\tilde{B}$ is the Brownian motion (not B). Also, we have defined the following quantities:

$$\mathcal{P} \coloneqq \sup_{t\in[0,T]}\mathbb{E}[\|P_t\|^2]\,, \qquad \mathcal{G} \coloneqq \sup_{t\in[0,T]}\mathbb{E}[\|\nabla V(X_t)\|^2]\,,$$

and

$$\Delta \coloneqq \sup_{t\in[nh,(n+1)h]}\mathbb{E}[\|s(X^{(K-1)}_{\tau(t)}) - \nabla V(X_t)\|^2]\,.$$

Applying Grönwall's inequality again,

$$\mathbb{E}[\|P^{(1)}_t\|^2] \lesssim \mathcal{P} + h^2\delta^2 + h^2\mathcal{G} + \gamma dh + h^2\Delta\,.$$

Substituting this into eq. (20),

$$\mathcal{E}_1 \lesssim h^2\mathcal{P} + h^4\delta^2 + h^4\mathcal{G} + \gamma dh^3 + h^4\Delta\,.$$

Substituting this into eq. (19) now yields

$$\mathcal{E}_K \lesssim \exp(-\Omega(K))\,(h^2\mathcal{P} + h^4\mathcal{G} + \gamma dh^3 + h^4\Delta) + \delta^2 h^4\,.$$

Recalling the definition of Δ and from eq. (16) and eq. (17), we have proven that

$$\begin{aligned}
\Delta \lesssim \delta^2 + \beta^2\Big(&\sup_{t\in[nh,(n+1)h]}\mathbb{E}[\|X_t - X_{\tau(t)}\|^2] \\
&+ \exp(-\Omega(K))\,(h^2\mathcal{P} + h^4\mathcal{G} + \gamma dh^3 + h^4\Delta) + \delta^2 h^4\Big)\,.
\end{aligned}$$

Using $h \lesssim 1/\sqrt{\beta}$, this yields

$$\Delta \lesssim \delta^2 + \beta^2 \Big(\sup_{t \in [nh,(n+1)h]} \mathbb{E}[\|X_t - X_{\tau(t)}\|^2] + \exp(-\Omega(K)) \left(h^2 \mathcal{P} + h^4 \mathcal{G} + \gamma d h^3\right) \Big) .$$

We also note that

$$\mathbb{E}[\|X_t - X_{\tau(t)}\|^2] = \mathbb{E}\Big[\Big\| \int_{\tau(t)}^t P_s \, ds \Big\|^2\Big] \leq \frac{h^2}{M^2} \mathcal{P} .$$

The quantities $\mathcal{P}, \mathcal{G}$ are controlled via Theorem 19. Now assume that $\exp(-\Omega(K)) \leq 1/M^4$, which only requires $K \gtrsim \log M$ for a sufficiently large absolute constant. When the dust settles,

$$\Delta \lesssim \delta^2 + \frac{\beta^2 d h^2}{M^2} + \frac{\beta^2 h^2}{M^2} \left(1 + \frac{\kappa}{M^2}\right) \mathsf{KL}(\mu_0 \parallel \pi)$$

Substituting this into eq. (15), and recalling that $\gamma \asymp \sqrt{\beta}$, we finally obtain

$$\mathsf{KL}(\pi_T \parallel \mu_T) \lesssim \frac{T}{\sqrt{\beta}} \Big(\delta^2 + \frac{\beta^2 d h^2}{M^2} + \frac{\beta^2 h^2}{M^2} \left(1 + \frac{\kappa}{M^2}\right) \mathsf{KL}(\mu_0 \parallel \pi) \Big) .$$

This completes the proof. ■

We must complement the discretization bound with a continuous-time convergence result, which can be obtained from off-the-shelf results. See Zhang et al. (2023, Lemma 5) for a statement which is convenient for our setting (adapted from Ma et al. (2021), which in turn followed the original entropic hypocoercivity due to Villani (Villani, 2009); see also Monmarché (2024) for the corresponding result for idealized Hamiltonian Monte Carlo).

Theorem 21 *Assume that V is β-smooth and that $\pi^X \propto \exp(-V)$ satisfies the LSI with constant α. Consider the functional*

$$\mathcal{F}(\mu \parallel \pi) := \mathsf{KL}(\mu \parallel \pi) + \mathbb{E}_\mu\big[\big\| \mathfrak{M}^{1/2} \nabla \log \frac{\mu}{\pi} \big\|^2\big] , \qquad \mathfrak{M} := \begin{bmatrix} 1/(4\beta) & 1/\sqrt{2\beta} \\ 1/\sqrt{2\beta} & 4 \end{bmatrix} \otimes I .$$

Then, for all $t \geq 0$,

$$\mathcal{F}(\pi_t \parallel \pi) \leq \exp\Big(-\frac{\alpha t}{10\sqrt{2\beta}}\Big) \mathcal{F}(\pi_0 \parallel \pi) .$$

We are now ready to prove Theorem 15.

Proof [Proof of Theorem 15] Let us show that $\mu_0 = \mathcal{N}(x^\star, \beta^{-1} I) \otimes \mathcal{N}(0, I)$ satisfies

$$\mathcal{F}(\mu_0 \parallel \pi) \leq \frac{d}{2} (2 + \log \kappa) .$$

From Theorem 14, we know that $\mathsf{KL}(\mu_0 \parallel \pi) \leq \frac{d}{2} \log \kappa$. Also,

$$\mathbb{E}_{\mu_0}\big[\big\| \mathfrak{M}^{1/2} \nabla \log \frac{\mu_0}{\pi} \big\|^2\big] = \frac{1}{4\beta} \mathbb{E}_{\mathcal{N}(x^\star, \beta^{-1} I)}\big[\big\| \nabla \log \frac{\mathcal{N}(x^\star, \beta^{-1} I)}{\pi^X} \big\|^2\big]$$

$$
\begin{aligned}
&= \frac{1}{4\beta}\,\mathbb{E}_{x\sim\mathcal{N}(x^\star,\beta^{-1}I)}\big[\big\|\nabla V(x) - \frac{\beta}{2}\,(x-x^\star)\big\|^2\big] \\
&\le \frac{1}{2\beta}\,\mathbb{E}_{x\sim\mathcal{N}(x^\star,\beta^{-1}I)}\big[\|\nabla V(x)-\nabla V(x^\star)\|^2 + \frac{\beta^2}{4}\,\|x-x^\star\|^2\big] \\
&\le \beta\,\mathbb{E}_{x\sim\mathcal{N}(x^\star,\beta^{-1}I)}[\|x-x^\star\|^2] \le d\,.
\end{aligned}
$$

The initialization bound follows.

The setting of parameters is such that from Theorem 20 and Theorem 21 respectively, we have $\mathsf{KL}(\pi_{Nh} \parallel \mu_{Nh}) \lesssim \varepsilon^2$ and $\mathsf{KL}(\pi_{Nh} \parallel \pi) \lesssim \varepsilon^2$. The result now follows from Pinsker's inequality and the triangle inequality for TV. ■

Appendix C. Proofs for sampling from discrete distributions

We begin with the proof of Theorem 16.

Proof [Proof of Theorem 16] The first two statements are from Anari et al. (2023). We only need to verify the last statement. We only need to show that we can approximate $\mathrm{mean}(\tau_z\mu)$ for all $z\in\mathbb{R}^n$, given the oracle for the Laplace transform of μ. Since μ is supported on the hypercube, we can rewrite the j-th entry of $\mathrm{mean}(\tau_z\mu)$ in term of Laplace transforms of μ, i.e.,

$$
\begin{aligned}
(\mathrm{mean}(\tau_z\mu))_j = 2\,\tau_z\mu(x_j=+) - 1 &= \frac{2\sum_{x\in\{\pm\}^n,\,x_j=+}\exp(\langle z,x\rangle)\,\mu(x)}{\sum_{x\in\{\pm\}^n}\exp(\langle z,x\rangle)\,\mu(x)} - 1 \\
&= \frac{2\exp(z_j)\sum_{x\in\{\pm\}^n,\,x_j=+}\exp(\langle z_{-j},x_{-j}\rangle)\,\mu(x)}{\sum_{x\in\{\pm\}^n}\exp(\langle z,x\rangle)\,\mu(x)} - 1 \\
&= 2\exp\big(z_j + \mathcal{L}_\mu z^+ - \mathcal{L}_\mu z\big) - 1\,,
\end{aligned}
$$

where z^+ (resp. z^-) is a vector with all entries equal to z except for the j-th entry being $+\infty$ (resp. $-\infty$). Using the oracle, we can compute $\hat{A}_+$ s.t. $|\hat{A}_+ - (\mathcal{L}_\mu z^+ - \mathcal{L}_\mu z)| \le O(\varepsilon)$. Thus,

$$
\begin{aligned}
&|2\exp(z_j+\hat{A}_+) - 1 - (\mathrm{mean}(\tau_z\mu))_j| \\
&\quad= 2\exp(z_j)\exp(\mathcal{L}_\mu z^+ - \mathcal{L}_\mu z)\,\big|\exp\big(\hat{A}_+ - (\mathcal{L}_\mu z^+ - \mathcal{L}_\mu z)\big) - 1\big| \\
&\quad\le O(\varepsilon)\exp(z_j)\exp(\mathcal{L}_\mu z^+ - \mathcal{L}_\mu z) = O(\varepsilon)\,\frac{(\mathrm{mean}(\tau_z\mu))_j + 1}{2} = O(\varepsilon)
\end{aligned}
$$

where the inequality follows from $\exp(x) - 1 \le 2x$ for $x\in[0,1/2)$. We use n machines, each of which computes one entry of $\mathrm{mean}(\tau_z\mu)$ using 2 oracle calls and $O(1)$ parallel iterations. The estimated score function s satisfies $\|s(y) - \nabla V(y)\| \lesssim \sqrt{\frac{n}{c}\,\varepsilon^2} = \delta$. ■

We also need another initialization lemma, since Theorem 14 requires knowledge of the minimizer of V which is not necessarily the case for the present application.

Lemma 22 *Let $\mu_0 = \mathcal{N}(y,\sigma^2 I)$ for some fixed $y\in\mathbb{R}^n$ and $\sigma^2 > 0$. If $\pi \propto \exp(-V)$ with $\nabla^2 V \preceq \beta I$, then*

$$
\mathsf{KL}(\mu_0 \parallel \pi) \le V(y) + \log Z + \frac{n}{2}\,(\beta\sigma^2 - \log(2\pi e\sigma^2))
$$

where $Z = \int \exp(-V(x))\,dx$.

Proof By smoothness, $V(x) \le V(y) + \langle \nabla V(y), x - y\rangle + \frac{\beta}{2}\|x-y\|^2$, thus

$$\mathbb{E}_{x\sim\mu_0} V(x) \le V(y) + \langle \nabla V(y), \mathbb{E}_{x\sim\mu_0} x - y\rangle + \frac{\beta}{2}\mathbb{E}_{x\sim\mu_0}\|x-y\|^2 = V(y) + \frac{\beta\sigma^2 n}{2}$$

and

$$\mathsf{KL}(\mu_0 \parallel \pi) = \mathbb{E}_{x\sim\mu_0} \log \mu_0(x) + V(x) + \log Z = -\frac{n}{2}\log(2\pi e\sigma^2) + V(y) + \frac{\beta\sigma^2 n}{2} + \log Z\,,$$

which is the desired bound. ■

Lemma 23 *Consider a density function $\nu : \{\pm 1\}^n \to \mathbb{R}_{\ge 0}$. Let $\pi = \nu * \mathcal{N}(0, cI)$ and $\mu_0 = \mathcal{N}(0, cI)$. Then,*

$$\mathsf{KL}(\mu_0 \parallel \pi) \le \frac{n}{2c}\,.$$

Proof We can write

$$\pi(y) = (2\pi c)^{-n/2} \sum_{x\in\{\pm1\}^n} \nu(x)\exp\big(-\frac{\|y-x\|^2}{2c}\big)\,.$$

This distribution is normalized so that $Z = 1$, and

$$\pi(0) = (2\pi c)^{-n/2} \sum_{x\in\{\pm1\}^n} \nu(x)\exp\big(-\frac{n}{2c}\big) = (2\pi c)^{-n/2}\exp\big(-\frac{n}{2c}\big)\,.$$

Thus, $V(0) = -\log\pi(0) = \frac{n}{2}\log(2\pi c) + \frac{n}{2c}$. By Theorem 16, $\nabla^2 V \preceq I/c$. Thus, we can apply theorem 22 with $\beta = c^{-1}$ and $\sigma^2 = c$. Rearranging gives the desired inequality. ■

Proof [Proof of Theorem 4] Let c be such that $\operatorname{cov}\tau_y\mu \preceq \frac{c}{2}I$ for all $y \in \mathbb{R}^n$. Suppose we have two executions of algorithm 3: one using the approximate continuous sampling algorithm resulting in $w_0, \dots, w_T$, and one using exact samples resulting in $w'_0, \dots, w'_T$. Note that $w_i = w_{i-1} + x_i/c$ where x_i is the output of Algorithm 1 on input $\pi = \tau_{w_{i-1}}\mu * \mathcal{N}(0, cI)$ and $w'_i = w'_{i-1} + x'_i/c$ where $x'_i \sim \tau_{w'_{i-1}}\mu * \mathcal{N}(0, cI)$. We choose the parameter of Algorithm 1 so that

$$\mathsf{TV}(\operatorname{law}(x_i), \tau_{w_{i-1}}\mu * \mathcal{N}(0, cI)) \le \eta$$

for some η to be specified later.

Recall that the total variation distance is also characterized as the smallest probability of error when we couple two random variables according to the two measures, i.e.,

$$\mathsf{TV}(\rho_1, \rho_2) = \inf\big\{\Pi(X_1 \ne X_2) \;\big|\; \Pi \text{ is a coupling of } (\rho_1, \rho_2)\big\}\,.$$

On the first iteration, we can couple x_1 with x'_1 so that they are equal to each other with probability at least $1 - \eta$. If $x_1 = x'_1$, then $w_1 = w'_1$, and repeating the argument on this event we can couple x_2 to x'_2 so that $x_2 = x'_2$ with probability at least $1 - \eta$. After T iterations, by the union bound, we have $w_T = w'_T$ with probability at least $1 - T\eta$.

By triangle inequality, the data-processing inequality, and Theorem 17,

$$\begin{aligned}\mathsf{TV}(\mathrm{law}(\operatorname{sign} w_T), \mu) &\le \mathsf{TV}(\mathrm{law}(\operatorname{sign} w_T), \mathrm{law}(\operatorname{sign} w'_T)) + \mathsf{TV}(\mathrm{law}(\operatorname{sign} w'_T), \mu) \\ &\le T\eta + \varepsilon/2\,,\end{aligned}$$

provided we choose $T = \Theta(c \log(n/\varepsilon))$ so that $\mathsf{TV}(\mathrm{law}(\operatorname{sign} w'_T), \mu) \le \varepsilon/2$. We then choose $\eta = \varepsilon/(2T)$, which ensures that $\mathsf{TV}(\mathrm{law}(\operatorname{sign} w_T), \mu) \le \varepsilon$.

In each iteration of the "for" loop in Algorithm 3, we want to approximately sample from $\pi = \tau_{w'_{i-1}}\mu * \mathcal{N}(0, cI)$, which is $(2c)^{-1}$-strongly log concave and c^{-1}-log-smooth by Theorem 16. By Theorem 23, $\mathsf{KL}(\mu_0 \parallel \pi) \le \mathrm{poly}(n)$ for $\mu_0 = \mathcal{N}(0, cI)$. Thus, by Theorem 13, to sample x'_i such that $\mathsf{TV}(\mathrm{law}(x'_i), \tau_{w'_{i-1}}\mu * \mathcal{N}(0, cI)) \le O(\varepsilon/(c \log(n/\varepsilon)))$, Algorithm 1 uses $P = O(\log^2(cn/\varepsilon))$ parallel iterations, $M = \widetilde{O}(c^2 n/\varepsilon^2)$ processors, and $MP = \widetilde{O}(c^2 n/\varepsilon^2)$ δ-approximate gradient evaluations with $\delta = \Theta(\varepsilon/\sqrt{c})$. By Theorem 16, each gradient evaluation can be implemented using $O(n)$ processors, $O(1)$ parallel iterations, and $O(n)$ total calls to $O(\delta\sqrt{c}/n) = O(\varepsilon/n)$-approximate Laplace transform oracles.

Hence, Algorithm 3 takes $PT = O(c \log^3(cn/\epsilon))$ parallel iterations, $M = \widetilde{O}(c^2 n^2/\varepsilon^2)$ processors, and $\widetilde{O}(c^2 n^2/\varepsilon^2)$ total calls to $O(\varepsilon/n)$-approximate Laplace transform oracles. ■