
Parallel Simulation for Log-concave Sampling and Score-based Diffusion Models

Huanjian Zhou^{1 2} Masashi Sugiyama^{2 1}

Abstract

Sampling from high-dimensional probability distributions is fundamental in machine learning and statistics. As datasets grow larger, computational efficiency becomes increasingly important, particularly in reducing *adaptive complexity*, namely the number of sequential rounds required for sampling algorithms. While recent works have introduced several parallelizable techniques, they often exhibit suboptimal convergence rates and remain significantly weaker than the latest lower bounds for log-concave sampling. To address this, we propose a novel parallel sampling method that improves adaptive complexity dependence on dimension d reducing it from $\tilde{O}(\log^2 d)$ to $\tilde{O}(\log d)$. Our approach builds on parallel simulation techniques from scientific computing.

1. Introduction

We study the problem of sampling from a probability distribution with density $\pi(x) \propto \exp(-f(x))$ where $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is a smooth potential. We consider two types of setting. **Problem (a):** the distribution is known only up to a normalizing constant (Chewi, 2023), and this kind of problem is fundamental in many fields such as Bayesian inference, randomized algorithms, and machine learning (Robert et al., 1999; Marin et al., 2007; Nakajima et al., 2019). **Problem (b):** known as the score-based generative models (SGMs) (Song & Ermon, 2019), we are given an approximation of $\nabla \log \pi_t$, where π_t is the density of a specific process at time t . The law of this process converges to π over time. SGMs are state-of-the-art in applications like image generation (Ho et al., 2022a; Dhariwal & Nichol, 2021), audio and video generation (Ho et al., 2022b; Yang et al., 2023a), and inverse problems (Song et al., 2022).

For Problem (a), specifically log-concave sampling, starting from the seminal papers of Dalalyan & Tsybakov (2012), Dalalyan (2017), and Durmus & Moulines (2017), there has been a flurry of recent works on proving non-asymptotic guarantees based on simulating a process which converges to π over time (Wibisono, 2018; Vempala & Wibisono, 2019; Mou et al., 2021; Altschuler & Talwar, 2023). Moreover, these processes, such as Langevin dynamics, converge exponentially quickly to π under mild conditions (Dalalyan, 2017; Mou et al., 2021; Bernard et al., 2022). Such dynamics-based algorithms for Problem (a) share a common feature with the inference process of SGMs that they are actually a numerical simulation of an initial-value problem of differential equations (Hodgkinson et al., 2021). Thanks to the exponentially fast convergence of the process, significant efforts have been conducted on discretizing these processes using numerical methods such as the forward Euler, backward Euler (proximal method), exponential integrator, mid-point, and high-order Runge-Kutta methods (Vempala & Wibisono, 2019; Wibisono, 2019; Shen & Lee, 2019; Li et al., 2019; Oliva & Akyildiz, 2024).

Furthermore, in recent years, there have been increasing interest and significant advances in understanding the convergence of inherently dynamics-based SGMs (Bortoli, 2022; Chen et al., 2023c; Lee et al., 2023; Pedrotti et al., 2024; Chen et al., 2023b; Tang & Zhao, 2024; Li & Yan, 2024). Notably, polynomial-time convergence guarantees have been established (Chen et al., 2023c;b; Benton et al., 2024; Liang et al., 2025), and various discretization schemes for SGMs have been analyzed (Lu et al., 2022a;b; Huang et al., 2025).

The algorithms underlying the above results are highly sequential. However, with the increasing size of data sets for sampling, we need to develop a theory for algorithms with limited iterations. For example, the widely-used denoising diffusion probabilistic models (Ho et al., 2020) may take 1000 denoising steps to generate one sample, while the evaluations of a neural network-based score function can be computationally expensive (Song et al., 2021).

As a comparison, recently, the (naturally parallelizable) Picard methods for diffusion models reduced the number of steps to around 50 (Shih et al., 2024). Furthermore, in terms

¹The University of Tokyo, Tokyo, Japan ²Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan. Correspondence to: Huanjian Zhou <zhou@ms.k.u-tokyo.ac.jp>.

of the dependency on the dimension d and accuracy ε , Picard methods for both Problems (a) and (b) were proven to be able to return an ε -accurate solution within $\mathcal{O}(\log^2(d/\varepsilon^2))$ iterations, improved from previous $\mathcal{O}(d^a/\varepsilon^b)$ with some $a, b > 0$. However, the $\mathcal{O}(\log^2(d/\varepsilon^2))$ adaptive complexity¹ may not be yet optimal for both Problem (a) and (b). This motivates our investigation into the question:

Can we achieve logarithmic adaptive complexity for both log-concave sampling and sampling for SGMs?

Our Contributions

In this work, we propose a novel sampling method that employs a highly parallel discretization approach for continuous processes, with applications to the overdamped Langevin diffusion (Chewi, 2023) and the stochastic differential equation (SDE) implementation of processes in SGMs (Chen et al., 2024) for Problems (a) and (b), respectively.

Faster parallel log-concave sampling. We first present an improved result for parallel sampling from a strongly log-concave and log-smooth distribution. Specifically, we improve the upper bound from $\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$ (Anari et al., 2024) to $\tilde{\mathcal{O}}(\log(\frac{d}{\varepsilon^2}))$, with slightly scaling the number of processors and gradient evaluations from $\mathcal{O}(\frac{d}{\varepsilon^2})$ to $\mathcal{O}(\frac{d}{\varepsilon^2} \log(\frac{d}{\varepsilon^2}))$.

Compared with methods based on underdamped Langevin diffusion (Shen & Lee, 2019; Yu & Dalalyan, 2024; Anari et al., 2024), our method exhibits higher space complexity². This is primarily because underdamped Langevin diffusion typically follows a smoother trajectory than overdamped Langevin diffusion, allowing for larger grid spacing and consequently, a reduced number of grids. We summarize the comparison in Table 1. In this paper, we will focus on the adaptive complexity and discretization schemes for overdamped Langevin diffusion.

Faster parallel sampling for diffusion models. We then present an improved result for diffusion models. Specifically, we propose an efficient algorithm with $\tilde{\mathcal{O}}(\log(\frac{d}{\varepsilon^2}))$ adaptive complexity for SDE implementations of diffusion models (Song & Ermon, 2019). Our method surpasses all the existing parallel methods for diffusion models having $\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$ adaptive complexity (Chen et al., 2024;

¹Adaptive complexity refers to the minimal number of sequential rounds required for an algorithm to achieve a desired accuracy, assuming polynomially many queries can be executed in parallel at each round (Balkanski & Singer, 2018).

²We note, in this paper, that the space complexity refers to the number of words (Cohen-Addad et al., 2023; Chen et al., 2024) instead of the number of bits (Goldreich, 2008) to denote the approximate required storage.

Table 1. Comparison with existing parallel methods for strongly log-concave sampling. The symbol \spadesuit represents that the results hold under a weaker condition, the log-Sobolev inequality.

Work dynamics	Measure	Adaptive Complexity	Space Complexity
(Shen & Lee, 2019, Theorem 4) underdamped Langevin diffusion	W_2	$\tilde{\mathcal{O}}(\log^2(\frac{\sqrt{d}}{\varepsilon}))$	$\tilde{\mathcal{O}}(\frac{d^{3/2}}{\varepsilon})$
(Yu & Dalalyan, 2024, Corollary 2) underdamped Langevin diffusion	W_2	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^{3/2}}{\varepsilon})$
(Anari et al., 2024, Theorem 15) underdamped Langevin diffusion	TV	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))^\spadesuit$	$\tilde{\mathcal{O}}(\frac{d^{3/2}}{\varepsilon})$
(Anari et al., 2024, Theorem 13) overdamped Langevin diffusion	KL	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))^\spadesuit$	$\tilde{\mathcal{O}}(\frac{d^2}{\varepsilon^2})$
Theorem 4.2 overdamped Langevin diffusion	KL	$\tilde{\mathcal{O}}(\log(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^2}{\varepsilon^2})$

Table 2. Comparison with existing parallel methods for sampling for diffusion models.

Works Implementation	Measure	Adaptive Complexity	Space Complexity
(Chen et al., 2024, Theorem 3.5) ODE / Picard method	TV	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^{3/2}}{\varepsilon^2})$
(Gupta et al., 2025, Theorem B.13) ODE / Parallel midpoint method	TV	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^{3/2}}{\varepsilon^2})$
(Chen et al., 2024, Theorem 3.3) SDE / Picard method	KL	$\tilde{\mathcal{O}}(\log^2(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^2}{\varepsilon^2})$
Theorem 5.4 SDE / Parallel Picard method	KL	$\tilde{\mathcal{O}}(\log(\frac{d}{\varepsilon^2}))$	$\tilde{\mathcal{O}}(\frac{d^2}{\varepsilon^2})$

Gupta et al., 2025), with slightly increasing the number of the processors and gradient evaluations and the space complexity for SDEs. We summarize the comparison in Table 2. Similarly, the better space complexity of the ordinary differential equation (ODE) implementations is attributed to the smoother trajectories of ODEs, which are more readily discretized.

2. Problem Set-up

In this section, we introduce some preliminaries and key ingredients of log-concave sampling and diffusion models in Sections 2.1 and 2.2, respectively. Following this, Section 2.3 provides an introduction to the fundamentals of Picard iterations.

2.1. Log-concave Sampling

Problem (a) (Sampling task). Given the potential function $f : \mathcal{D} \rightarrow \mathbb{R}$, the goal of the sampling task is to draw a sample from the density $\pi_f = Z_f^{-1} \exp(-f)$, where $Z_f := \int_{\mathcal{D}} \exp(-f(x)) dx$ is the normalizing constant.

Distribution and function class. If f is (strongly) convex, the density π_f is said to be (strongly) *log-concave*. If f is twice-differentiable and $\nabla^2 f \preceq \beta \mathbf{I}$ (where \preceq denotes the Loewner order and \mathbf{I} is the identity matrix), we say the potential f is β -smooth and the density π_f is β -log-smooth.

We define *relative Fisher information* of probability density ρ w.r.t. π as $\text{FI}(\rho|\pi) = \mathbb{E}_\rho[\|\nabla \log(\rho/\pi)\|^2]$ and the *Kullback-Leibler (KL) divergence* of ρ from π as $\text{KL}(\rho|\pi) = \mathbb{E}_\rho \log(\rho/\pi)$. If π is α -strongly log-concave, then the following relation between KL divergence and Fisher informa-

tion holds:

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

Langevin Dynamics. One of the most commonly-used dynamics for sampling is Langevin dynamics (Chewi, 2023), which is the solution to the following SDE,

$$d\mathbf{x} = -\nabla f(\mathbf{x})dt + \sqrt{2}d\mathbf{B}_t,$$

where $(\mathbf{B}_t)_{t \in [0, T]}$ is a standard Brownian motion in \mathbb{R}^d . If $\pi \propto \exp(-f)$ satisfies strongly log-concavity, then the law of the Langevin diffusion converges exponentially fast to π (Bakry et al., 2014).

Score function for sampling task. We assume the score function $\mathbf{s} : \mathbb{R}^d \rightarrow \mathbb{R}$ is a pointwise accurate estimate of ∇f , i.e., $\|\mathbf{s}(\mathbf{x}) - \nabla f(\mathbf{x})\| \leq \delta$ for all $\mathbf{x} \in \mathbb{R}^d$ and some sufficiently small constant $\delta \in \mathbb{R}_+$.

Measures of the output. For two densities ρ and π , we define the *total variation* (TV) as

$$\text{TV}(\rho, \pi) = \sup\{\rho(E) - \pi(E) \mid E \text{ is an event}\}.$$

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

$$\text{TV}(\rho, \pi) \leq \sqrt{\frac{1}{2} \text{KL}(\rho\|\pi)}.$$

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

$$W_2^2(\rho, \pi) = \inf_{\Pi} \mathbb{E}_{(X, Y) \sim \Pi} [\|X - Y\|^2],$$

where the infimum is over coupling distributions Π of (X, Y) such that $X \sim \rho$, $Y \sim \pi$. If π is α -strongly log-concave, the following transport-entropy inequality, known as Talagrand's T_2 inequality, holds (Otto & Villani, 2000) for all $\rho \in \mathcal{P}_2(\mathbb{R}^d)$, i.e., with finite second moment,

$$\frac{\alpha}{2} W_2^2(\rho, \pi) \leq \text{KL}(\rho\|\pi).$$

Complexity. For any sampling algorithm, we consider the *adaptive complexity* defined as unparallelizable evaluations of the score function (Chen et al., 2024), and use the notion of the *space complexity* to denote the approximate required storage during the inference. We note, in this paper, that the space complexity refers to the number of words (Cohen-Addad et al., 2023; Chen et al., 2024) instead of the number of bits (Goldreich, 2008) to denote the approximate required storage.

2.2. Score-based Diffusion Models

Sampling for diffusion models. In score-based diffusion models, one considers forward process $(\mathbf{x}_t)_{t \in [0, T]}$ in \mathbb{R}^d governed by the canonical Ornstein-Uhlenbeck (OU) process (Ledoux, 2000):

$$d\mathbf{x}_t = -\frac{1}{2}\mathbf{x}_t dt + d\mathbf{B}_t, \quad \mathbf{x}_0 \sim \mathbf{q}_0, \quad t \in [0, T], \quad (1)$$

where \mathbf{q}_0 is the initial distribution over \mathbb{R}^d . The corresponding backward process $(\tilde{\mathbf{x}}_t)_{t \in [0, T]}$ in \mathbb{R}^d follows an SDE defined as

$$\begin{cases} d\tilde{\mathbf{x}}_t = [\frac{1}{2}\tilde{\mathbf{x}}_t + \nabla \log \tilde{p}_t(\tilde{\mathbf{x}}_t)] dt + d\mathbf{B}_t & t \in [0, T], \\ \tilde{\mathbf{x}}_0 \sim \mathbf{p}_0 \approx \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d) \end{cases} \quad (2)$$

where $\mathcal{N}(\cdot, \cdot)$ represents the normal distribution over \mathbb{R}^d . In practice, the score function $\nabla \log \tilde{p}_t(\tilde{\mathbf{x}}_t)$ is estimated by neural network (NN) $\mathbf{s}_t^\theta : \mathbb{R}^d \mapsto \mathbb{R}^d$, where θ is the parameters of NN. The backward process is approximated by

$$\begin{cases} d\mathbf{y}_t = [\frac{1}{2}\mathbf{y}_t + \mathbf{s}_t^\theta(\mathbf{y}_t)] dt + d\mathbf{B}_t & t \in [0, T], \\ \mathbf{y}_0 \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d). \end{cases} \quad (3)$$

Problem (b) (Sampling task for SGMs). Given the learned NN-based score function \mathbf{s}_t^θ , the goal is to simulate the approximated backward process such that the law of the output is close to \mathbf{q}_0 .

Distribution class. For SGMs, we assume the data density p_0 has finite second moments and is normalized such that $\text{cov}_{p_0}(\mathbf{x}_0) = \mathbb{E}_{p_0}[(\mathbf{x}_0 - \mathbb{E}_{p_0}[\mathbf{x}_0])(\mathbf{x}_0 - \mathbb{E}_{p_0}[\mathbf{x}_0])^\top] = \mathbf{I}_d$. Such a finite moment assumption is standard across previous theoretical works on SGMs (Chen et al., 2023a;b;c) and we adopt the normalization to simplify true score function-related computations as Benton et al. (2024) and Chen et al. (2024) did.

OU process and inverse process The OU process and its inverse process also converge to the target distribution exponentially fast in various divergences and metrics such as the 2-Wasserstein metric W_2 ; see Ledoux (2000). Furthermore, the discrepancy between the terminal distributions of the backward process (2) and its approximation version (3) scales polynomially with respect to the length of the time horizon and the score matching error. (Huang et al. (2025, Theorem 3.5) or setting the step size $h \rightarrow 0$ for the results in Chen et al. (2023a;b;c)).

Score function for SGMs. For the NN-based score, we assume the score function is L^2 -accurate, bounded and Lipschitz; we defer the details in Section 5.2.

2.3. Picard Iterations

Consider the integral form of the initial value problem, $\mathbf{x}_t = \mathbf{x}_0 + \int_0^t f_s(\mathbf{x}_s)ds + \sqrt{2}\mathbf{B}_t$. The main idea of Picard iterations (Clenshaw, 1957) is to approximate the difference over time slice $[t_n, t_{n+1}]$ as

$$\begin{aligned} & \mathbf{x}_{t_{n+1}} - \mathbf{x}_{t_n} \\ &= \int_{t_n}^{t_{n+1}} f_s(\mathbf{x}_s)ds + \sqrt{2}(\mathbf{B}_{t_{n+1}} - \mathbf{B}_{t_n}) \\ &\approx \sum_{i=1}^M \mathbf{w}_i \int_{t_n+\tau_{n,i}}^{t_{n+1}} f_s(\mathbf{x}_{t_n+\tau_{n,i}})ds + \sqrt{2}(\mathbf{B}_{t_{n+1}} - \mathbf{B}_{t_n}), \end{aligned}$$

with a discrete grids of M collocation points as $\mathbf{x}_{t_n} = \mathbf{x}_{t_n+\tau_{n,0}} \leq \mathbf{x}_{t_n+\tau_{n,1}} \leq \mathbf{x}_{t_n+\tau_{n,2}} \leq \dots \leq \mathbf{x}_{t_n+\tau_{n,M}} = \mathbf{x}_{t_{n+1}}$. We update the points in a wave-like fashion, which inherently allows for parallelization: for $m' = 1, \dots, M$,

$$\begin{aligned} \mathbf{x}_{t_n+\tau_{n,m}}^{p+1} &= \mathbf{x}_0 + \sum_{m'=1}^{m-1} \mathbf{w}_{m'} f_{t_n+\tau_{n,m'}}(\mathbf{x}_{t_n+\tau_{n,m'}}^p) \\ &\quad + \sqrt{2}(\mathbf{B}_{t_n+\tau_{n,m}} - \mathbf{B}_{t_n}). \end{aligned}$$

Various collocation points have been proposed, including uniform points and Chebyshev points (Bai & Junkins, 2011). In this paper, however, we focus exclusively on the simplest case of uniform points, and extension to other cases is future work. Picard iterations are known to converge exponentially fast and, under certain conditions, even factorially fast for ODEs and backward SDEs (Hutzenthaler et al., 2021).

3. Technical Overview

We adopt the time splitting for the time horizon used in the existing parallel methods (Shen & Lee, 2019; Gupta et al., 2025; Chen et al., 2024; Anari et al., 2024; Yu & Dalalyan, 2024). With same time grids, our algorithm, however, depart crucially from prior work in the design of parallelism across the time slices, and the modification for controlling the score estimation error. Below we summarize these algorithmic contributions and technical novelties.

Recap of existing parallel sampling methods. Existing works for parallel sampling apply the following generic discretization schemes (Shen & Lee, 2019; Gupta et al., 2025; Chen et al., 2024; Anari et al., 2024; Yu & Dalalyan, 2024). At a high level, these methods divide the time horizon into many large time slices and each slice is further subdivided into grids with a small enough step size. Instead of sequentially updating the grid points, they update all grids at the same time slice simultaneously using exponentially fast converging Picard iterations (Alexander, 1990; Chen et al., 2024; Anari et al., 2024), or randomized midpoint methods (Shen & Lee, 2019; Yu & Dalalyan, 2024; Gupta

et al., 2025). With $\tilde{\mathcal{O}}(\log d)$ Picard iterations for $\tilde{\mathcal{O}}(\log d)$ time slices, the total adaptive complexity of their algorithms is $\tilde{\mathcal{O}}(\log^2 d)$. However, while sequential updating of each time slice is not necessary for simulating the process, it remains unclear how to parallelize across time slices for sampling to obtain $\mathcal{O}(\log d)$ time complexity.

Algorithmic novelty: parallel methods across time slices.

Naïvely, if we directly update all the grids simultaneously, the Picard iterations will not converge when the total length is $T = \tilde{\mathcal{O}}(\log d)$. Instead of updating all time slices together or updating the time slice sequentially, we update the time slices in a *diagonal* style as illustrated in Figure 1. For any j -th update at the n -th time slice (corresponding the rectangle in the n -th column from the left and the j -th row from the top in Figure 1), there will be two inputs: (a) the right boundary point of the previous time slice, which has been updated j times, and (b) the points on the grids of the same time slice that have been updated $j - 1$ times. Then we perform P times Picard iterations with these inputs, where the hyperparameter P depends on the smoothness of the score function. This repetition is simple but essential for preventing the accumulation of score-matching errors, which could otherwise grow exponentially w.r.t. the length of the time horizon. The main difference compared to the existing Picard methods is that for a fixed time slice, the starting points in our method are updated gradually, whereas in existing methods, the starting points remain fixed once processed.

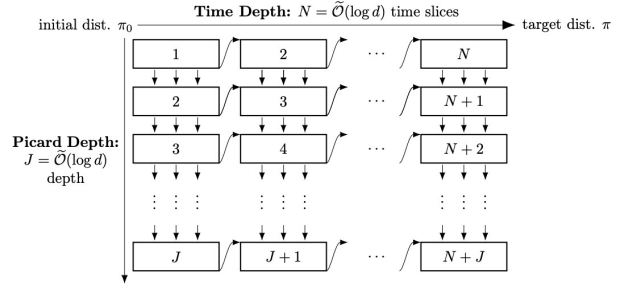


Figure 1. Illustration of the parallel Picard method: each rectangle represents an update, and the number within each rectangle indicates the index of the Picard iteration. The approximate time complexity is $N + J = \tilde{\mathcal{O}}(\log d)$.

Challenges for convergence. Similar to the arguments for sequential methods or parallel methods with sequentially updating the time slices, we use the standard techniques such as the interpolation method or Girsanov’s theorem (Vempala & Wibisono, 2019; Oksendal, 2013; Chewi, 2023) to decompose the total error w.r.t. KL into four components: (i) convergence error of the continuous process, (ii) discretization error, (iii) parallelization error, and (iv) score estimation error (See Eq. (4), Lemma B.1, and Lemma C.4). For (i) the convergence error of the continuous processes,

their exponential convergence rates allow this error to be effectively controlled by setting the total time length to $\mathcal{O}(\log \frac{d}{\varepsilon^2})$, regardless of the specific discretization scheme. For (ii), the discretization error scales approximately as $d \cdot \frac{h}{M}$, where h denotes the time slice length, M the number of discretization points per slice, and $\frac{h}{M}$ the grid resolution (See Eq. (4), and Lemma C.5). Setting $\frac{h}{M} \approx \mathcal{O}(\varepsilon^2/d)$ ensures the discretization error remains within $\mathcal{O}(\varepsilon^2)$. The technical challenges rise from controlling the remaining two errors, which we summarize below.

(iii) *Parallelization error*: the parallelization error primarily arises from updating with $s(x^{j-1})$ instead of $s(x^j)$ in the Picard iteration, in contrast to the sequential method, where j indexes the steps along the Picard direction. In existing parallel methods, the sequential update across time slices benefits the convergence of truncation errors, $\mathbb{E}[\|x^j - x^{j-1}\|^2]$. Assuming the truncation errors in the previous time slice have converged, its right boundary serves as the starting point for all grids in the current $\mathcal{O}(1)$ -length time slice which results in an initial bias of $\mathcal{O}(d)$. Subsequently, by performing $\mathcal{O}(\log d)$ exponentially fast Picard iterations, the truncation error will converge. However, in our diagonal-style updating scheme across time, the truncation error interacts with inputs from both the previous time slice and prior updates in the same time slice. Consequently, the bias-convergence loop that holds in sequential updating no longer holds.

(iv) *Score estimation error*: If the score function itself is Lipschitz continuous (Assumption 5.3 for Problem (b)), no additional score matching error will arise during the Picard iterations. This allows the total score estimation error to remain bounded under mild conditions (Assumption 5.1). However, for Problem (a), since it is the velocity field ∇f instead of the score function s that is Lipschitz, additional score estimation errors will occur during each update. For the sequential algorithm, these additional score estimation errors are contained within the bias-convergence loop, ensuring the total score estimation error remains to be bounded. Conversely, for our diagonal-style updating algorithm, the absence of convergence along the time direction causes these additional score estimation errors to accumulate exponentially over the time direction.

Technical novelty. Our technical contributions address these challenges by the appropriate selection of the number of Picard iterations within each update P and the depth of the Picard iterations J . We outline the details of the choices below.

In the following, we assume that the truncation error at the n -th time slice and the j -th iteration scales with L_n^j , and that the additional score estimation error for each update scales with δ^2 .

To address the initial challenge related to the truncation error, we choose the Picard depth as $J = \mathcal{O}(N + \log d)$. We first bound the error of the output for each update with respect to its inputs as $L_n^j \leq aL_{n-1}^j + bL_n^{j-1}$, where a and b are constants. By carefully choosing the length of the time slices, we can ensure that $b < 1$ along the Picard iteration direction. Consequently, the truncation error will converge if the iteration depth J is sufficiently large, such that $a^N b^J$ is sufficiently small. This requirement implies that $J = \mathcal{O}(N + \log d)$.

To mitigate the additional score estimation error for Problem (a), we perform P Picard iterations within each update. The interaction between the truncation error and additional score estimation error can be expressed as $L_n^j \leq aL_{n-1}^j + bL_n^{j-1} + c\delta^2$, where a, b, c are constants. To ensure the total score estimation error remains bounded, it is necessary to have $a, b < 1$, which guarantees convergence along both the time and Picard directions. By the convergence of the Picard iteration, we can achieve $b < 1$. For a , the right boundary point of the previous time slice, and prior updates within the same time slice introduce discrepancies in the truncation error. For the impact from the previous time slice, we make use of the contraction of gradient decent to ensure convergence. However, since the grid gap scale as $1/d$, the contraction factor is close to 1. Consequently, we have to minimize the impact from prior updates within the same time slice, which scales as $\mathcal{O}(1)$ by repeating $P = \log \mathcal{O}(1)$ Picard iterations for each update.

Balance between time and Picard directions. We note that the Picard method, despite being the simplest approach for time parallelism, has achieved the state of art performance in certain specific settings. On the one hand, the continuous processes need to run for at least $\mathcal{O}(\log d)$ time. To ensure convergence within every time slice, the time slice length have to be set as $\mathcal{O}(1)$, resulting in a necessity for at least $\mathcal{O}(\log d)$ iterations. On the other hand, with a proper initialization $\mathcal{O}(d)$, Picard iterations converge within $\mathcal{O}(\log d)$ iterations. Our parallelization balances the convergence of the continuous diffusion and the Picard iterations to achieve the improved results.

Related works in scientific computation. Similar parallelism across time slices has also been proposed in scientific computation (Gear, 1991; Gander, 2015; Ong & Schroder, 2020), especially for parallel Picard iterations (Wang, 2023). Compared with prior work in scientific computation, our approach exhibits several significant differences. Firstly, our primary objective differs from that in simulation. In sampling, we aim to ensure that the output distribution closely approximates the target distribution, whereas simulation seeks to make some points on the discrete grid closely match the true dynamics. Second, our algorithm differs sig-

nificantly from that of Wang (2023). In our algorithm, each update takes the inputs without the corrector operation. Furthermore, we perform P Picard iterations in each update to prevent error accumulation over time $T = \tilde{O}(\log d)$. However, these two fields are connected through the sampling strategies that ensure each discrete point closely approximates the true process at every sampling step.

4. Parallel Picard Method for Strongly Log-concave Sampling

In this section, we present parallel Picard methods for strongly log-concave sampling (Algorithm 1) and show it holds improved convergence rate w.r.t. the KL divergence and total variance (Theorem 4.2 and Corollary 4.3). We illustrate the algorithm in Section 4.1, and give a proof sketch in Section 4.3. All the missing proofs can be found in Appendix B.

4.1. Algorithm

Our parallel Picard method for strongly log-concave sampling is summarized in Algorithm 1. In Lines 2–7, we generate the noises and initialize the value at the grid via Langevin Monte Carlo (Chewi, 2023) with a stepsize $h = \mathcal{O}(1)$. In Lines 8–26, the time slices are updated in a diagonal manner within the outer loop, as illustrated in Figure 1. In Lines 12–14 and Lines 21–23, we repeat P Picard iterations for each update to ensure convergence.

Remark 4.1. Parallelization should be understood as evaluating the score function concurrently, with each time slice potentially being computed in an asynchronous parallel manner, resulting in the overall $P(N + J) + N$ adaptive complexity.

4.2. Theoretical Guarantees

The following theorem summarizes our theoretical analysis for Algorithm 1.

Theorem 4.2. *Suppose π is α -strongly log-concave and β -log-smooth, and the score function s is δ -accurate. Let $\kappa = \beta/\alpha$. Suppose*

$$\beta h = 0.1, \quad M \geq \frac{\kappa d}{\varepsilon^2}, \quad N \geq 10\kappa \log \left(\frac{\text{KL}(\mu_0 \parallel \pi)}{\varepsilon^2} \right),$$

$$\delta \leq 0.2\sqrt{\alpha}\varepsilon, \quad P \geq \frac{2 \log \kappa}{3} + 4,$$

$$\text{and } J - N \geq \log \left(N^3 \left(\frac{\kappa \delta^2 h + \kappa \text{KL}(\mu_0 \parallel \pi) + \kappa^2 d}{\varepsilon^2} \right) \right).$$

then Algorithm 1 runs within $N + (N + J)P$ iterations with at most MN queries per iteration and outputs a sample with marginal distribution ρ such that

$$\max \left\{ \frac{\sqrt{\alpha}}{2} W_2(\rho, \pi), \text{TV}(\rho, \pi) \right\} \leq \sqrt{\frac{\text{KL}(\rho, \pi)}{2}} \leq 2\varepsilon.$$

Algorithm 1 Parallel Picard Method for sampling

```

1: Input:  $x_0 \sim \mu_0$ , approximate score function  $s \approx \nabla f$ ,
   the number of the iterations in outer loop  $J$ , the number
   of the iteration in inner loop  $P$ , the number of time
   slices  $N$ , the length of time slices  $h$ , the number of
   points on each time slices  $M$ .
2: for  $n = 0, \dots, N - 1$  do
3:   for  $m = 0, \dots, M$  (in parallel) do
4:      $B_{nh+m/Mh} = B_{nh} + \mathcal{N}(0, (mh/M)\mathbf{I}_d)$ ,
5:      $x_{-1,M}^j = x_0$ , for  $j = 0, \dots, J$ ,
6:      $x_{n,m}^0 = x_{n-1,M}^0 - \frac{hm}{M} s(x_{n-1,M}^0) + \sqrt{2}(B_{nh+mh/M} - B_{nh})$ ,
7:   end for
8: end for
9: for  $k = 1, \dots, N$  do
10:  for  $j = 1, \dots, \min\{k - 1, J\}$  and  $m = 1, \dots, M$ 
    (in parallel) do
11:    let  $n = k - j$ ,  $x_{n,0}^j = x_{n-1,M}^j$ , and  $x_{n,m}^{j,0} = x_{n,m}^{j-1}$ ,
12:    for  $p = 1, \dots, P$  do
13:       $x_{n,m}^{j,p} = x_{n,0}^j - \frac{h}{M} \sum_{m'=0}^{m-1} s(x_{n,m'}^{j,p-1}) + \sqrt{2}(B_{nh+mh/M} - B_{nh})$ ,
14:    end for
15:     $x_{n,m}^j = x_{n,m}^{j,P}$ ,
16:  end for
17: end for
18: for  $k = N + 1, \dots, N + J - 1$  do
19:  for  $n = \max\{0, k - J\}, \dots, N - 1$  and  $m = 1, \dots, M$ 
    (in parallel) do
20:    let  $j = k - n$ ,  $x_{n,0}^j = x_{n-1,M}^j$ , and  $x_{n,m}^{j,0} = x_{n,m}^{j-1}$ ,
21:    for  $p = 1, \dots, P$  do
22:       $x_{n,m}^{j,p} = x_{n,0}^j - \frac{h}{M} \sum_{m'=0}^{m-1} s(x_{n,m'}^{j,p-1}) + \sqrt{2}(B_{nh+mh/M} - B_{nh})$ ,
23:    end for
24:     $x_{n,m}^j = x_{n,m}^{j,P}$ ,
25:  end for
26: end for
27: Return:  $x_{N-1,M}^J$ .
    
```

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

Corollary 4.3. *Suppose that $\pi = \exp(-f)$ is α -strongly log-concave and β -log-smooth, and let $\kappa = \beta/\alpha$. Let x^* be the minimizer of f . Then, for $\mu_0 = \mathcal{N}(x^*, \beta^{-1})$, it holds that $\text{KL}(\mu_0 \parallel \pi) \leq \frac{d}{2} \log \kappa$. Consequently, setting*

$$h = \frac{1}{10\beta}, \quad M = \frac{\kappa d}{\varepsilon^2}, \quad N = 10\kappa \log \left(\frac{d \log \kappa}{\varepsilon^2} \right),$$

$$\delta \leq 0.2\sqrt{\alpha}\varepsilon, \quad P \geq \frac{2 \log \kappa}{3} + 4,$$

$$\text{and } J - N = \mathcal{O}\left(\log \frac{\kappa^2 d \log \kappa}{\varepsilon^2}\right),$$

then Algorithm 1 runs within $N + (N + J)P = \tilde{\mathcal{O}}(\kappa \log \frac{d}{\varepsilon^2})$ iterations with at most $MN = \tilde{\mathcal{O}}(\frac{\kappa^2 d}{\varepsilon^2} \log \frac{d}{\varepsilon^2})$ queries per iteration and outputs a sample with marginal distribution ρ such that

$$\max \left\{ \frac{\sqrt{\alpha}}{2} W_2(\rho, \pi), \text{TV}(\rho, \pi) \right\} \leq \sqrt{\frac{\text{KL}(\rho, \pi)}{2}} \leq 2\varepsilon.$$

Remark 4.4. The main drawback of our method is the sub-optimal space complexity due to its application to overdamped Langevin diffusion which has a less smooth trajectory compared to underdamped Langevin diffusion. However, we anticipate that our method could achieve comparable space complexity when adapted to underdamped Langevin diffusion.

Remark 4.5. Regarding the condition number κ , our method achieves the same adaptive complexity of $\mathcal{O}(\kappa)$ as both the state-of-the-art sequential method and existing parallel approaches (Anari et al., 2024; Yu & Dalalyan, 2024; Altschuler & Chewi, 2024). Whether parallelization can improve the dependence on the condition number remains an open question, which we leave for future work.

Remark 4.6. When the number of computation cores, denoted by ℓ , is limited, the adaptive complexity of our algorithm is $\tilde{\mathcal{O}}(\frac{\kappa^2 d}{\varepsilon^2 \ell} \log^2 \frac{d}{\varepsilon^2})$. This matches the state-of-the-art adaptive complexity of Anari et al. (2024) and Yu & Dalalyan (2024) when $\ell \leq \frac{\kappa d}{\varepsilon^2}$. However, our algorithm achieves improved adaptive complexity when $\ell \geq \frac{\kappa d}{\varepsilon^2}$, with potential applications as demonstrated in Nishihara et al. (2014), De Souza et al. (2022), Hafych et al. (2022) and Glatt-Holtz et al. (2024).

4.3. Proof Sketch of Theorem 4.2: Performance Analysis of Algorithm 1

The detailed proof of Theorem 4.2 is deferred to Appendix B. As discussed in Section 3, by interpolation methods (Anari et al., 2024), we decompose the error w.r.t. the KL divergence into four error components (corollary B.4):

$$\begin{aligned} & \text{KL}(\rho \parallel \pi) \\ & \lesssim \underbrace{e^{-\Theta(N)} \text{KL}(\mu_0 \parallel \pi)}_{\text{Convergence of Langevin dynamics}} + \underbrace{\frac{dh}{M}}_{\text{discretization error}} \\ & + \underbrace{\sum_{n=1}^{N-1} e^{-\Theta(n)} \mathcal{E}_{N-n}^J}_{\text{parallization error}} + \underbrace{\delta^2}_{\text{score estimation error}}, \end{aligned} \quad (4)$$

where \mathcal{E}_n^j represents the truncation error of the grids at n -th time slice after j update. For the right terms, with the choice of $N = \mathcal{O}(\log \frac{d}{\varepsilon^2})$, $M = \mathcal{O}(dh/\varepsilon^2)$ and $\delta \leq \varepsilon$, which

ensures a sufficiently long time horizon the sufficiently long time horizon $T = Nh = \mathcal{O}(\log \frac{d}{\varepsilon^2})$, densely spaced grids with a gap $h/M = \mathcal{O}(\varepsilon^2/d)$ and a small score matching error, respectively, we can conclude that

$$e^{-\Theta(N)} \text{KL}(\mu_0 \parallel \pi) + \frac{dh}{M} + \delta^2 \lesssim \varepsilon^2$$

Thus, we will focus on proving the convergence of the truncation error \mathcal{E}_n^j in the Picard iterations, and avoiding the additional accumulation of the score estimation error during Picard iterations as discussed before.

Considering that the truncation error expands at most exponentially along the time direction, but diminishes exponentially with an increased depth of the Picard iterations, convergence can be achieved by ensuring that the depth of the Picard iterations surpasses the number of time slices as $J \geq N + \mathcal{O}(\log \frac{d}{\varepsilon^2})$ with initialization error bounded by $\mathcal{O}(d)$ (the second part of Corollary B.7 and second part of Corollary B.9).

Due to the non-Lipschitzness of the score function, we can only bound \mathcal{E}_n^j by quantity $a\Delta_{n-1}^j + b\mathcal{E}_n^{j-1} + c\delta^2 h^2$ (Lemma B.5 and Lemma B.8), where Δ_{n-1}^j represents the truncation error at the right boundary of the previous time slice. Here, the coefficients are given by: $a = 1 - 0.1 \frac{\beta h}{\varepsilon} + \mathcal{O}(\kappa)(3\beta^2 h^2)^P$, $b = \mathcal{O}((\beta^2 h^2)^P)$ and $c = \mathcal{O}(\kappa \delta^{\frac{d}{\kappa}} h^2)$. Intuitively, a comes from the contraction of the gradient mapping with an additional term from the Picard direction, b reflects convergence along the Picard direction, and c accounts for the accumulation of score estimation error δ over time length h , with an additional scaling by κ due to Young's inequality. To control the growth of the score error, it is essential that the coefficients a and b remain strictly less than one. Setting $P = \Theta(\log \kappa)$ is sufficient to ensure this condition.

5. Parallel Picard Method for Sampling of Diffusion Models

In this section, we present parallel Picard methods for diffusion models in Section 5.1 and assumptions in Section 5.2. Then we show it holds improved convergence rate w.r.t. the KL divergence (Theorem 5.4). All the missing details can be found in Appendix C.

5.1. Algorithm

Due to space limitations, the detailed methodology for the parallelization of Picard methods for diffusion models is provided in Appendix C.1 and Algorithm 2. It keeps same parallel structure as that illustrated in Figure 1. Notably, it exhibits the following distinctions in comparison to the parallel Picard methods for strongly log-concave sampling presented in Algorithm 1:

- Since the score function itself is Lipschitz, there will not be additional score matching error during Picard iterations. As a result, we perform single Picard iteration in one update, i.e., $P = 1$;
- Instead of uniform discrete grids, we employ a shrinking step size discretization scheme towards the data end, and the early stopping technique which is unavoidable to show the convergence for diffusion models (Chen et al., 2024). We show the details in Appendix C.1;
- We use an exponential integrator instead of the Euler-Maruyama Integrator in Picard iterations, where an additional high-order discretization error term would emerge (Chen et al., 2023a), which we believe would not affect the overall $\mathcal{O}(\log d)$ adaptive complexity with parallel sampling.

5.2. Assumptions

Our theoretical analysis of the algorithm assumes mild conditions regarding the data distribution’s regularity and the approximation properties of NNs as discussed in Chen et al. (2024). These assumptions align with those established in previous theoretical works, such as those described by Chen et al. (2023c;a;b; 2024).

Assumption 5.1 ($(L^2([0, t_N]) \delta_2$ -accurate learned score). The learned NN-based score s_t^θ is δ_2 -accurate in the sense of

$$\mathbb{E}_{\tilde{p}} \left[\sum_{n=0}^{N-1} \sum_{m=0}^{M_n-1} \epsilon_{n,m} \left\| s_{t_n+\tau_{n,m}}^\theta(\tilde{x}_{t_n+\tau_{n,m}}) - \nabla \log \tilde{p}_{t_n+\tau_{n,m}}(\tilde{x}_{t_n+\tau_{n,m}}) \right\|^2 \right] \leq \delta_2^2.$$

Assumption 5.2 (Regular and normalized data distribution). The data density p_0 has finite second moments and is normalized such that $\text{cov}_{p_0}(x_0) = I_d$.

Assumption 5.3 (Bounded and Lipschitz learned NN-based score). The learned NN-based score function s_t^θ has a bounded \mathcal{C}^1 norm, i.e., $\|s_t^\theta(\cdot)\|_{L^\infty([0,T])} \leq M_s$ with Lipschitz constant L_s .

5.3. Theoretical Guarantees

Theorem 5.4. Under Assumptions 5.1, 5.2, and 5.3, given the following choices of the order of the parameters

$$h = \Theta(1), \quad N = \mathcal{O}\left(\log \frac{d}{\varepsilon^2}\right), \quad M = \mathcal{O}\left(\frac{d}{\varepsilon^2} \log \frac{d}{\varepsilon^2}\right),$$

$$T = \mathcal{O}\left(\log \frac{d}{\varepsilon^2}\right), \quad \delta \leq \varepsilon, \quad \text{and} \quad J = \mathcal{O}\left(N + \log \frac{Nd}{\varepsilon^2}\right)$$

the parallel Picard algorithm for diffusion models (Algorithm 2) generates samples from satisfies the following error bound,

$$\text{KL}(p_\eta \| \tilde{q}_{t_N}) \lesssim de^{-T} + \frac{dT}{M} + \varepsilon^2 + \delta_2^2 \lesssim \varepsilon^2, \quad (5)$$

with total $2N + J = \tilde{\mathcal{O}}\left(\log \frac{d}{\varepsilon^2}\right)$ adaptive complexity and $dM = \tilde{\mathcal{O}}\left(\frac{d^2}{\varepsilon^2}\right)$ space complexity for parallelizable δ_2 -accurate score function computations.

Remark 5.5. Compared to existing parallel methods, our method improves the adaptive complexity from $\mathcal{O}(\log^2 \frac{d}{\varepsilon^2})$ to $\mathcal{O}(\log \frac{d}{\varepsilon^2})$. Its main drawback is suboptimal space complexity due to the less smooth trajectory in SDE implementations, but we believe it can achieve comparable space complexity when adapted to ODE implementations.

Remark 5.6. When the number of computation cores ℓ is limited, the adaptive complexity is $\tilde{\mathcal{O}}\left(\frac{\kappa^2 d}{\varepsilon^2 \ell} \log^2 \frac{d}{\varepsilon^2}\right)$, matching the state-of-the-art result of Chen et al. (2024) for $\ell \leq \frac{\kappa d}{\varepsilon^2}$. However, employing a large batch size $\ell \geq \frac{\kappa d}{\varepsilon^2}$ in diffusion models may not always yield substantial benefits (Shih et al., 2024; Li et al., 2024b;a).

Remark 5.7. We note that the uniformly Lipschitz assumption (Assumption 5.3) may be too strong. In particular, the required Lipschitz constant can become quite large even becoming unbounded near the zero point (Salmona et al., 2022; Yang et al., 2023b). In this case, to ensure convergence of the Picard iterations under these conditions, the quantity $L_s^2 e^{h_n} h_n$ must be sufficiently small. This requirement implies that the length of each time slice, h_n , should scale as $\mathcal{O}(1/L_s^2)$. Consequently, the number of time slices becomes $N = \mathcal{O}(L_s^2 \log d)$, leading to an overall iteration complexity of $N = \mathcal{O}(L_s^2 \log d)$. We also believe our algorithm, which is based on a diagonal-style update, is robust to this assumption by adaptively adjusting the length of the time slices.

6. Discussion and Conclusion

In this work, we proposed novel parallel Picard methods for various sampling tasks. Notably, we obtain ε^2 -accurate sample w.r.t. the KL divergence within $\tilde{\mathcal{O}}\left(\log \frac{d}{\varepsilon^2}\right)$, which represents a significant improvement from $\tilde{\mathcal{O}}\left(\log^2 \frac{d}{\varepsilon^2}\right)$ for diffusion models. Furthermore compared with the existing methods applied to the overdamped Langevin dynamics or the SDE implementations for diffusion models, our space complexity only scales by a logarithmic factor.

Our study opens several promising theoretical directions. First, as an analogue to simulation methods in scientific computing, it highlights the potential of leveraging alternative discretization techniques for faster and more efficient sampling. Another direction is exploring smoother dynamics to reduce space complexity in these methods.

Lastly, although our highly parallel methods may introduce engineering challenges, such as the memory bandwidth, we believe our theoretical works will motivates the empirical development of parallel algorithms for both sampling and diffusion models.

Acknowledgment

The authors thank Sinho Chewi for very helpful conversations. HZ was supported by International Graduate Program of Innovation for Intelligent World and Next Generation Artificial Intelligence Research Center. MS was supported by JST ASPIRE Grant Number JPMJAP2405.

Impact Statement

This work focuses on the theory of accelerating the sampling via parallelism. As far as we can see, there is no foreseeable negative impact on the society.

References

- Alexander, R. Solving Ordinary Differential Equations i: Nonstiff Problems. *Siam Review*, 1990.
- Altschuler, J. and Talwar, K. Resolving the Mixing Time of the Langevin Algorithm to its Stationary Distribution for Log-Concave Sampling. In *Proceedings of Thirty Sixth Conference on Learning Theory*, 2023.
- Altschuler, J. M. and Chewi, S. Shifted Composition III: Local Error Framework for KL Divergence. *arXiv preprint arXiv:2412.17997*, 2024.
- Anari, N., Chewi, S., and Vuong, T.-D. Fast parallel sampling under isoperimetry. *Proceedings of Thirty Seventh Conference on Learning Theory*, pp. 161–185, 2024.
- Bai, X. and Junkins, J. L. Modified Chebyshev-Picard iteration methods for orbit propagation. *The Journal of the Astronautical Sciences*, 2011.
- Bakry, D., Gentil, I., Ledoux, M., et al. *Analysis and geometry of Markov diffusion operators*, volume 103. Springer, 2014.
- Balkanski, E. and Singer, Y. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th annual ACM SIGACT Symposium on Theory of Computing*, 2018.
- Benton, J., Bortoli, V. D., Doucet, A., and Deligiannidis, G. Nearly d -Linear Convergence Bounds for Diffusion Models via Stochastic Localization. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.
- Bernard, É., Fathi, M., Levitt, A., and Stoltz, G. Hypocoercivity with schur complements. *Annales Henri Lebesgue*, 5:523–557, 2022.
- Bortoli, V. D. Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.
- Chen, H., Lee, H., and Lu, J. Improved Analysis of Score-based Generative Modeling: User-Friendly Bounds under Minimal Smoothness Assumptions. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4735–4763. PMLR, 2023a.
- Chen, H., Ren, Y., Ying, L., and Rotskoff, G. M. Accelerating Diffusion Models with Parallel Sampling: Inference at Sub-Linear Time Complexity. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Chen, S., Chewi, S., Lee, H., Li, Y., Lu, J., and Salim, A. The probability flow ODE is provably fast. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b.
- Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023c.
- Chewi, S. Log-concave sampling. *Book draft available at <https://chewisinho.github.io>*, 2023.
- Chewi, S., Erdogdu, M. A., Li, M., Shen, R., and Zhang, M. S. Analysis of Langevin Monte Carlo from Poincaré to Log-Sobolev. *Foundations of Computational Mathematics*, 2024.
- Clenshaw, C. The numerical solution of linear differential equations in Chebyshev series. In *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press, 1957.
- Cohen-Addad, V., Woodruff, D. P., and Zhou, S. Streaming Euclidean k -median and k -means with $o(\log n)$ Space. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023.
- Dalalyan, A. S. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2017.
- Dalalyan, A. S. and Tsybakov, A. B. Sparse regression learning by aggregation and Langevin Monte-Carlo. *Journal of Computer and System Sciences*, 2012.

- De Souza, D. A., Mesquita, D., Kaski, S., and Acerbi, L. Parallel MCMC without embarrassing failures. In *International Conference on Artificial Intelligence and Statistics*, pp. 1786–1804. PMLR, 2022.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- Dragomir, S. S. Some Gronwall type inequalities and applications. *Science Direct Working Paper*, 2003.
- Durmus, A. and Moulines, E. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. 2017.
- Dwivedi, R., Chen, Y., Wainwright, M. J., and Yu, B. Log-concave sampling: Metropolis-Hastings algorithms are fast. *Journal of Machine Learning Research*, 2019.
- Gander, M. J. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg, May 6-8, 2013*, pp. 69–113. Springer, 2015.
- Gear, C. Waveform methods for space and time parallelism. *Journal of Computational and Applied Mathematics*, 1991.
- Glatt-Holtz, N. E., Holbrook, A. J., Krometis, J. A., and Mondaini, C. F. Parallel MCMC algorithms: theoretical foundations, algorithm design, case studies. *Transactions of Mathematics and Its Applications*, 8(2):tnae004, 2024.
- Goldreich, O. Computational complexity: a conceptual perspective. *ACM Sigact News*, 2008.
- Gupta, S., Cai, L., and Chen, S. Faster Diffusion-based Sampling with Randomized Midpoints: Sequential and Parallel. *International Conference on Learning Representations*, 2025.
- Hafych, V., Eller, P., Schulz, O., and Caldwell, A. Parallelizing MCMC sampling via space partitioning. *Statistics and Computing*, 32(4):56, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.
- Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022a.
- Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b.
- Hodgkinson, L., Salomone, R., and Roosta, F. Implicit Langevin algorithms for sampling from log-concave densities. *Journal of Machine Learning Research*, 22(136): 1–30, 2021.
- Huang, D. Z., Huang, J., and Lin, Z. Convergence Analysis of Probability Flow ODE for Score-Based Generative Models. *IEEE Transactions on Information Theory*, 71(6):4581–4601, 2025.
- Hutzenthaler, M., Kruse, T., and Nguyen, T. A. On the speed of convergence of Picard iterations of backward stochastic differential equations. *arXiv preprint arXiv:2107.01840*, 2021.
- Ledoux, M. The geometry of markov diffusion generators. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, 2000.
- Lee, H., Lu, J., and Tan, Y. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*. PMLR, 2023.
- Li, G. and Yan, Y. Adapting to unknown low-dimensional structures in score-based diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Li, M., Cai, T., Cao, J., Zhang, Q., Cai, H., Bai, J., Jia, Y., Li, K., and Han, S. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7183–7193, 2024a.
- Li, S., Yang, L., Jiang, X., Lu, H., Di, Z., Lu, W., Chen, J., Liu, K., Yu, Y., Lan, T., et al. SwiftDiffusion: Efficient Diffusion Model Serving with Add-on Modules. *arXiv preprint arXiv:2407.02031*, 2024b.
- Li, X., Wu, Y., Mackey, L., and Erdogdu, M. A. Stochastic Runge-Kutta Accelerates Langevin Monte Carlo and Beyond. *Advances in neural information processing systems*, 32, 2019.
- Liang, Y., Ju, P., Liang, Y., and Shroff, N. Broadening Target Distributions for Accelerated Diffusion Models via a Novel Analysis Approach. *The Thirteenth International Conference on Learning Representations, ICLR*, 2025.
- Lu, C., Zhou, Y., Bao, F., Chen, J., and Li, C. A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *Proc. Adv. Neural Inf. Process. Syst., New Orleans, United States*, pp. 1–31, 2022a.
- Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

- Marin, J.-M., Robert, C. P., et al. *Bayesian Core: A Practical Approach to Computational Bayesian statistics*, volume 268. Springer, 2007.
- Mou, W., Ma, Y.-A., Wainwright, M. J., Bartlett, P. L., and Jordan, M. I. High-order Langevin diffusion yields an accelerated MCMC algorithm. *Journal of Machine Learning Research*, 2021.
- Nakajima, S., Watanabe, K., and Sugiyama, M. *Variational Bayesian learning theory*. Cambridge University Press, 2019.
- Nishihara, R., Murray, I., and Adams, R. P. Parallel MCMC with generalized elliptical slice sampling. *The Journal of Machine Learning Research*, 15(1):2087–2112, 2014.
- Oksendal, B. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.
- Oliva, P. F. V. and Akyildiz, O. D. Kinetic Interacting Particle Langevin Monte Carlo. *arXiv preprint arXiv:2407.05790*, 2024.
- Ong, B. W. and Schroder, J. B. Applications of time parallelization. *Computing and Visualization in Science*, 23: 1–15, 2020.
- Otto, F. and Villani, C. Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality. *Journal of Functional Analysis*, 2000.
- Pedrotti, F., Maas, J., and Mondelli, M. Improved Convergence of Score-Based Diffusion Models via Prediction-Correction. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.
- Robert, C. P., Casella, G., and Casella, G. *Monte Carlo statistical methods*, volume 2. Springer, 1999.
- Salmona, A., De Bortoli, V., Delon, J., and Desolneux, A. Can push-forward generative models fit multimodal distributions? *Advances in Neural Information Processing Systems*, 35:10766–10779, 2022.
- Shen, R. and Lee, Y. T. The Randomized Midpoint Method for Log-Concave Sampling. *Advances in Neural Information Processing Systems*, 32, 2019.
- Shih, A., Belkhale, S., Ermon, S., Sadigh, D., and Anari, N. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.
- Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.
- Tang, W. and Zhao, H. Contractive diffusion probabilistic models. *arXiv preprint arXiv:2401.13115*, 2024.
- Vempala, S. S. and Wibisono, A. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. In *Advances in Neural Information Processing Systems 32*, pp. 8092–8104, 2019.
- Wang, Y. Parallel Numerical Picard Iteration Methods. *J. Sci. Comput.*, 95(1):27, 2023. doi: 10.1007/S10915-023-02156-Y. URL <https://doi.org/10.1007/s10915-023-02156-y>.
- Wibisono, A. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*, 2018.
- Wibisono, A. Proximal Langevin algorithm: Rapid convergence under isoperimetry. *arXiv preprint arXiv:1911.01469*, 2019.
- Yang, R., Srivastava, P., and Mandt, S. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10):1469, 2023a.
- Yang, Z., Feng, R., Zhang, H., Shen, Y., Zhu, K., Huang, L., Zhang, Y., Liu, Y., Zhao, D., Zhou, J., et al. Lipschitz singularities in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023b.
- Yu, L. and Dalalyan, A. Parallelized Midpoint Randomization for Langevin Monte Carlo. *arXiv preprint arXiv:2402.14434*, 2024.
- Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023.

A. Useful tools

A.1. Girsanov's Theorem

Following the notation introduced in [Chen et al. \(2024, Appendix A.2\)](#), we consider a probability space (Ω, \mathcal{F}, p) on which $(\mathbf{w}_t(\omega))_{t \geq 0}$ is a Wiener process in \mathbb{R}^d , with the filtration $\{\mathcal{F}_t\}_{t \geq 0}$. For an Itô process $\mathbf{z}_t(\omega)$ satisfies $d\mathbf{z}_t(\omega) = \boldsymbol{\alpha}(t, \omega)dt + \boldsymbol{\Sigma}(t, \omega)d\mathbf{w}_t(\omega)$, we denote the marginal distribution of \mathbf{z}_t by p_t , and the path measure of the process \mathbf{z}_t by $p_{t_1:t_2}$.

Definition A.1. Assume \mathcal{B} is the Borel σ -algebra on \mathbb{R}^d . For any $0 \leq t_1 < t_2$, we define \mathcal{V} as the class of functions $f(t, \omega) : [0, +\infty) \times \Omega \rightarrow \mathbb{R}$ which is $\mathcal{B} \times \mathcal{F}_t$ measurable, \mathcal{F}_t -adapted for any $t \geq 0$ and satisfies

$$\mathbb{E} \left[\exp \left(\int_0^t f^2(t, \omega) dt \right) \right] < +\infty, \quad \forall t > 0.$$

For vectors and matrices, we say it belongs to \mathcal{V}^n or $\mathcal{V}^{m \times n}$ if each component of the vector or each entry of the matrix belongs to \mathcal{V} .

Theorem A.2 ((Chen et al., 2024, Corollary A.4)). Let $\boldsymbol{\alpha}(t, \omega) \in \mathcal{V}^m$, $\boldsymbol{\Sigma}(t, \omega) \in \mathcal{V}^{m \times n}$, and $(\mathbf{w}_t(\omega))_{t \geq 0}$ be a Wiener process on the probability space (Ω, \mathcal{F}, q) . For $t \in [0, T]$, suppose $\mathbf{z}_t(\omega)$ is an Itô process with the following SDE:

$$d\mathbf{z}_t(\omega) = \boldsymbol{\alpha}(t, \omega)dt + \boldsymbol{\Sigma}(t, \omega)d\mathbf{w}_t(\omega), \quad (6)$$

and there exist processes $\boldsymbol{\delta}(t, \omega) \in \mathcal{V}^n$ and $\boldsymbol{\beta}(t, \omega) \in \mathcal{V}^m$ such that:

1. $\boldsymbol{\Sigma}(t, \omega)\boldsymbol{\delta}(t, \omega) = \boldsymbol{\alpha}(t, \omega) - \boldsymbol{\beta}(t, \omega)$;
2. The process $M_t(\omega)$ as defined below is a martingale with respect to the filtration $\{\mathcal{F}_t\}_{t \geq 0}$ and probability measure q :

$$M_t(\omega) = \exp \left(- \int_0^t \boldsymbol{\delta}(s, \omega)^\top d\mathbf{w}_s(\omega) - \frac{1}{2} \int_0^t \|\boldsymbol{\delta}(s, \omega)\|^2 ds \right),$$

then there exists another probability measure p on (Ω, \mathcal{F}) such that:

1. $p \ll q$ with the Radon-Nikodym derivative $\frac{dp}{dq}(\omega) = M_T(\omega)$,
2. The process $\tilde{\mathbf{w}}_t(\omega)$ as defined below is a Wiener process on (Ω, \mathcal{F}, p) :

$$\tilde{\mathbf{w}}_t(\omega) = \mathbf{w}_t(\omega) + \int_0^t \boldsymbol{\delta}(s, \omega) ds,$$

3. Any continuous path in $\mathcal{C}([t_1, t_2], \mathbb{R}^m)$ generated by the process \mathbf{z}_t satisfies the following SDE under the probability measure p :

$$d\tilde{\mathbf{z}}_t(\omega) = \boldsymbol{\beta}(t, \omega)dt + \boldsymbol{\Sigma}(t, \omega)d\tilde{\mathbf{w}}_t(\omega). \quad (7)$$

Corollary A.3 ((Chen et al., 2024, Corollary A.5)). Suppose the conditions in Theorem A.2 hold, then for any $t_1, t_2 \in [0, T]$ with $t_1 < t_2$, the path measure of the SDE (7) under the probability measure p in the sense of $p_{t_1:t_2} = p(\mathbf{z}_{t_1:t_2}^{-1}(\cdot))$ is absolutely continuous with respect to the path measure of the SDE (6) in the sense of $q_{t_1:t_2} = q(\mathbf{z}_{t_1:t_2}^{-1}(\cdot))$. Moreover, the KL divergence between the two path measures is given by

$$\text{KL}(p_{t_1:t_2} \| q_{t_1:t_2}) = \text{KL}(p_{t_1} \| q_{t_1}) + \mathbb{E}_{\omega \sim p|_{\mathcal{F}_{t_1}}} \left[\frac{1}{2} \int_{t_1}^{t_2} \|\boldsymbol{\delta}(t, \omega)\|^2 dt \right].$$

A.2. Comparison Inequalities

Theorem A.4 (Gronwall inequality (Dragomir, 2003, Theorem 1)). Let x , Ψ and χ be real continuous functions defined in $[a, b]$, $\chi(t) \geq 0$ for $t \in [a, b]$. We suppose that on $[a, b]$ we have the inequality

$$x(t) \leq \Psi(t) + \int_a^t \chi(s)x(s)ds.$$

Then

$$x(t) \leq \Psi(t) + \int_a^t \chi(s)\Psi(s) \exp \left[\int_s^t \chi(u)du \right] ds.$$

A.3. Help Lemmas for diffusion models

Lemma A.5 ((Chen et al., 2023a, Lemma 9)). For $\hat{q}_0 \sim \mathcal{N}(0, I_d)$ and $\tilde{p} = p_T$ is the distribution of the solution to the forward process ((2)), we have

$$\text{KL}(\tilde{p}_0 \| \hat{q}_0) \lesssim de^{-T}.$$

B. Missing Proof for Log-concave Sampling

We denote $\text{KL}_n^j = \text{KL}(\mu_{n,M}^j \| \pi)$ where $\mu_{n,M}^j$ represents the law of $\mathbf{x}_{n,M}^j$. We define the truncation error from prior update as, $\mathcal{E}_n^j := \max_{m=1, \dots, M} \mathbb{E} [\|\mathbf{x}_{n,m}^{j,P} - \mathbf{x}_{n,m}^{j,P-1}\|^2]$, and truncation error at right boundary point as, $\Delta_n^j := \mathbb{E} [\|\mathbf{x}_{n,M}^j - \mathbf{x}_{n,M}^{j-1}\|^2]$.

B.1. One Step Analysis of KL_n^j : From KL's Convergence to Picard Convergence

In this section, we use the interpolation method to analyse the change of KL_n^j along time direction, which will be bounded by discretization error and score error.

Lemma B.1. Assume $\beta h \leq 0.1$. For any $j = 1, \dots, J$, $n = 1, \dots, N-1$, we have

$$\text{KL}_n^j \leq \exp(-1.2\alpha h) \text{KL}_{n-1}^j + \frac{0.5\beta dh}{M} + 4.4\beta^2 h \mathcal{E}_n^j + 2.1\delta^2 h.$$

Furthermore, for initialization part, i.e., $j = 0$, $n = 0, \dots, N-1$, we have

$$\text{KL}_n^0 \leq \exp(-\alpha(n+1)h) \text{KL}(\mu_0 \| \pi) + \frac{8\beta^2 dh}{\alpha},$$

Remark B.2. In the first equation, the term $\exp(-1.2\alpha h) \text{KL}_{n-1}^j$ characterizes the convergence of the continuous diffusion. Additionally, the second and third terms quantify the discretization error. Adopting $P = 0$ and $M = 1$ reverts to the classical scenario, where the discretization error approximates $\mathcal{O}(hd)$, as discussed in Section 4.1 of Chewi (2023). Moreover, the second term is influenced by the density of the grids, while the third term is dependent on the convergence of the Picard iterations. The fourth term accounts for the score error.

Proof. We will use the interpolation method and follow the proof of Theorem 13 in Anari et al. (2024). For $j \in [J]$, $n = 0, \dots, N-1$ and $m = 0, \dots, M-1$, it is easy to see that

$$\mathbf{x}_{n,m+1}^j = \mathbf{x}_{n,m}^j - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1}) + \sqrt{2}(B_{nh+(m+1)h} - B_{nh+mh/M}).$$

Let \mathbf{x}_t denote the linear interpolation between $\mathbf{x}_{n,m+1}^j$ and $\mathbf{x}_{n,m}^j$, i.e., for $t \in [nh + \frac{mh}{M}, nh + \frac{(m+1)hh}{M}]$, let

$$\mathbf{x}_t = \mathbf{x}_{n,m}^j - \left(t - nh - \frac{mh}{M}\right) \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1}) + \sqrt{2}(B_t - B_{nh+mh/M}).$$

Note that $\mathbf{s}(\mathbf{x}_{n,m}^{j,P})$ is a constant vector field. Let μ_t be the law of \mathbf{x}_t . The same argument as in Vempala & Wibisono (2019, Lemma 3/Equation 32) yields the differential inequality

$$\begin{aligned} \partial_t \text{KL}(\mu_t \| \pi) &= -\text{FI}(\mu_t \| \pi) + \mathbb{E} \left\langle \nabla f(\mathbf{x}_t) - \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1}), \nabla \log \frac{\mu_t(\mathbf{x}_t)}{\pi(\mathbf{x}_t)} \right\rangle \\ &\leq -\frac{3}{4} \text{FI}(\mu_t \| \pi) + \mathbb{E} [\|\nabla f(\mathbf{x}_t) - \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2], \end{aligned} \quad (8)$$

where we used $(a, b) \leq \frac{1}{4}\|a\|^2 + \|b\|^2$ and $\mathbb{E} [\|\nabla \log \frac{\mu_t(\mathbf{x}_t)}{\pi(\mathbf{x}_t)}\|^2] = \text{FI}(\mu_t \| \pi)$. For the first term, by α strongly-log-concavity of π , we have $\text{KL}(\mu_t \| \pi) \leq \frac{1}{2\alpha} \text{FI}(\mu_t \| \pi)$. For the second term, we have

$$\begin{aligned} &\mathbb{E} [\|\nabla f(\mathbf{x}_t) - \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2] \\ &\leq 2\mathbb{E} [\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{n,m}^{j,P-1})\|^2] + 2\mathbb{E} [\|\nabla f(\mathbf{x}_{n,m}^{j,P-1}) - \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2] \\ &\leq 2\beta^2 \mathbb{E} [\|\mathbf{x}_t - \mathbf{x}_{n,m}^{j,P-1}\|^2] + 2\delta^2. \end{aligned} \quad (9)$$

Moreover,

$$\mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] \leq 2\mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^j\|^2 \right] + 2\mathbb{E} \left[\|\mathbf{x}_{n,m}^{j,P} - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] \quad (10)$$

For the first term, which will be influenced by density of grids, we have

$$\begin{aligned} & \mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^j\|^2 \right] \\ & \leq \left(t - nh - \frac{mh}{M} \right)^2 \mathbb{E} \left[\|\mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2 \right] + d \left(t - nh - \frac{mh}{M} \right) \\ & \leq \frac{h^2}{M^2} \mathbb{E} \left[\|\mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2 \right] + d \left(t - nh - \frac{mh}{M} \right) \\ & \leq \frac{2h^2}{M^2} \mathbb{E} \left[\|\nabla f(\mathbf{x}_{n,m}^{j,P-1})\|^2 \right] + \frac{2\delta^2 h^2}{M^2} + \frac{dh}{M} \\ & \leq \frac{4\beta^2 h^2}{M^2} \mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] + \frac{4h^2}{M^2} \mathbb{E} \left[\|\nabla f(\mathbf{x}_t)\|^2 \right] + \frac{2\delta^2 h^2}{M^2} + \frac{dh}{M}. \end{aligned} \quad (11)$$

Taking $\beta h \leq \frac{1}{10}$, and combining (10) and (11), we have

$$\mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] \leq \frac{4.4h^2}{M^2} \mathbb{E} \left[\|\nabla f(\mathbf{x}_t)\|^2 \right] + \frac{2.2\delta^2 h^2}{M^2} + \frac{1.1dh}{M} + 2.2\mathbb{E} \left[\|\mathbf{x}_{n,m}^j - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right]. \quad (12)$$

For the first term, we recall the following lemma.

Lemma B.3 ((Chewi et al., 2024, Lemma 16)).

$$\mathbb{E} \left[\|\nabla f(\mathbf{x}_t)\|^2 \right] \leq \text{FI}(\mu_t \|\pi) + 2\beta d.$$

Combining (8), (9), (12) and $\beta h \leq \frac{1}{10}$, we have for $j \in [J]$, $n = 0, \dots, n-1$, $m = 0, \dots, M-1$, and $t \in \left[nh + \frac{mh}{M}, nh + \frac{(m+1)hh}{M} \right]$,

$$\begin{aligned} & \partial_t \text{KL}(\mu_t \|\pi) \\ & \leq -\frac{3}{4} \text{FI}(\mu_t \|\pi) + \mathbb{E} \left[\|\nabla f(\mathbf{x}_t) - \mathbf{s}(\mathbf{x}_{n,m}^{j,P-1})\|^2 \right] \\ & \leq -\frac{3}{4} \text{FI}(\mu_t \|\pi) + 2\beta^2 \mathbb{E} \left[\|\mathbf{x}_t - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] + 2\delta^2 \\ & \leq -\frac{3}{4} \text{FI}(\mu_t \|\pi) + \frac{8.8\beta^2 h^2}{M^2} \mathbb{E} \left[\|\nabla f(\mathbf{x}_t)\|^2 \right] + \frac{4.4\beta^2 \delta^2 h^2}{M^2} + \frac{2.2\beta^2 dh}{M} + 4.4\beta^2 \mathbb{E} \left[\|\mathbf{x}_{n,m}^{j,P} - \mathbf{x}_{n,m}^{j,P-1}\|^2 \right] + 2\delta^2 \\ & \leq -\frac{3}{4} \text{FI}(\mu_t \|\pi) + \frac{0.1}{M^2} \mathbb{E} \left[\|\nabla V(X_t)\|^2 \right] + \frac{0.1\delta^2}{M^2} + \frac{2.2\beta^2 dh}{M} + 4.4\beta^2 \mathcal{E}_n^j + 2\delta^2 \\ & \leq -\frac{3}{4} \text{FI}(\mu_t \|\pi) + \frac{0.1}{M^2} (\text{FI}(\mu_t \|\pi) + 2\beta d) + \frac{0.1\delta^2}{M^2} + \frac{2.2\beta^2 dh}{M} + 4.4\beta^2 \mathcal{E}_n^j + 2\delta^2 \\ & \leq -1.2\alpha \text{KL}(\mu_t \|\pi) + \frac{0.5\beta d}{M} + 4.4\beta^2 \mathcal{E}_n^j + 2.1\delta^2 \end{aligned}$$

Since this inequality holds independently of m , we integral from $t = nh$ to $t = (n+1)h$,

$$\text{KL}_n^j \leq \exp(-1.2\alpha h) \text{KL}_{n-1}^j + \frac{0.5\beta dh}{M} + 4.4\beta^2 h \mathcal{E}_n^j + 2.1\delta^2 h.$$

As for $j = 0$, actually, Line 4-7 performs a Langevin Monte Carlo with step size h , by Theorem 4.2.6 in Chewi (2023), we have

$$\text{KL}_n^0 \leq \exp(-\alpha nh) \text{KL}_0^0 + \frac{8dh\beta^2}{\alpha},$$

with $0 < h \leq \frac{1}{4L}$.

□

Corollary B.4. Assume $\beta h \leq 0.1$. We have

$$\text{KL}_{N-1}^J \leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2 h \Delta_0^J) + \sum_{n=1}^{N-1} e^{-1.2\alpha(n-1)h} 4.4\beta^2 h \mathcal{E}_{N-n}^J + \frac{0.5\beta d}{\alpha M} + \frac{2.1\delta^2}{\alpha}.$$

Furthermore, if \mathcal{E}_{N-n}^J has a uniform bound as $\mathcal{E}_{N-n}^J \leq \mathcal{E} + 500\delta^2 h^2$, we have

$$\text{KL}_{N-1}^J \leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2 h \Delta_0^J) + 5\beta\kappa\mathcal{E} + \frac{0.5\beta d}{\alpha M} + \frac{2.5\delta^2}{\alpha}.$$

Proof. By Lemma B.1, we decompose KL_{N-1}^J as

$$\begin{aligned} \text{KL}_{N-1}^J &\leq e^{-1.2\alpha(N-1)h} \text{KL}_0^J + \sum_{n=1}^{N-1} e^{-1.2\alpha(n-1)h} \left(\frac{0.5\beta d h}{M} + 4.4\beta^2 h \mathcal{E}_{N-n}^J + 2.1\delta^2 h \right) \\ &\leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2 h \Delta_0^J) \\ &\quad + \frac{4.4\beta^2 h (\mathcal{E} + 500\delta^2 h^2) + \frac{0.5\beta d h}{M} + 2.1\delta^2 h}{1 - \exp(-1.2\alpha h)} \\ &\leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2 h \Delta_0^J) + \frac{1.1}{\alpha h} 4.4\beta^2 h \mathcal{E} + \frac{1.1}{\alpha h} \frac{0.5\beta d h}{M} \\ &\quad + \frac{1.1}{\alpha h} 25\delta^2 h \\ &= e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2 h \Delta_0^J) + 5\kappa\beta\mathcal{E} + \frac{0.6\beta d}{\alpha M} + \frac{28\delta^2}{\alpha}, \end{aligned}$$

where the third inequality holds since $0 < x < 0.4$, we have $1.1 - 1.1 \exp(-1.2x) - x > 0$. It is clear that $\alpha h < \beta h < 0.1$. \square

B.2. One Step Analysis of Δ_n^j

In this section, we analyze the one step change of Δ_n^j first.

Lemma B.5. Assume $\beta h = \frac{1}{10}$ and $P \geq \frac{2\log \kappa}{3} + 4$. For any $j = 2, \dots, J$, $n = 1, \dots, N-1$, we have

$$\Delta_n^j \leq \left(1 - \frac{0.005}{\kappa}\right) \Delta_{n-1}^j + 4.4 \left(\frac{1}{M} + 10\kappa\right) h^2 \delta^2 + 4.4 \left(\frac{1}{M} + 10\kappa\right) \beta^2 h^2 \mathcal{E}_n^{j-1}.$$

Furthermore, for $j = 1$, $n = 1, \dots, N-1$, we have

$$\Delta_n^1 \leq \Delta_{n-1}^1 + \left(\frac{1}{M} + 10\kappa\right) \left(5\delta^2 h^2 + 6\beta^2 d h^3 + 0.4\beta^2 h^2 \frac{\text{KL}_{n-1}^0}{\alpha}\right).$$

Proof. **Decomposition when $j \geq 2$.** In fact, for $j \in [J]$, $n = 0, \dots, N-1$, $m = 0, \dots, M-1$, and $p = 1, \dots, P$, it is easy to see that

$$\mathbf{x}_{n,m+1}^{j,p} = \mathbf{x}_{n,m}^{j,p} - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m}^{j,p-1}) + \sqrt{2}(B_{nh+(m+1)h} - B_{nh+mh/M}).$$

For any $j = 2, \dots, J$, $n = 1, \dots, N-1$, by the contraction of $\phi(\mathbf{x}) = \mathbf{x} - \frac{h}{M} \nabla f(\mathbf{x})$ (Lemma 2.2 in Altschuler & Talwar (2023)), for any $m = 1, \dots, M$, we have,

$$\begin{aligned} &\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{j,P} - \mathbf{x}_{n,m}^{j-1,P} \right\|^2 \right] \\ &= \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m-1}^{j,P-1}) - \left(\mathbf{x}_{n,m-1}^{j-1,P} - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m-1}^{j-1,P-1}) \right) \right\|^2 \right] \end{aligned}$$

$$\begin{aligned}
 &\leq (1+\eta) \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j,P}) - \left(\mathbf{x}_{n,m-1}^{j-1,P} - \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j-1,P}) \right) \right\|^2 \right] \\
 &\quad + \left(2 + \frac{2}{\eta} \right) \mathbb{E} \left[\left\| \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j,P}) - \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j,P-1}) + \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j-1,P}) - \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j-1,P-1}) \right\|^2 \right] \\
 &\quad + \left(2 + \frac{2}{\eta} \right) \mathbb{E} \left[\left\| \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j,P-1}) - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m-1}^{j,P-1}) + \frac{h}{M} \nabla f(\mathbf{x}_{n,m-1}^{j-1,P-1}) - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m-1}^{j-1,P-1}) \right\|^2 \right] \\
 &\leq (1+\eta) \left(1 - \frac{\alpha h}{M} \right)^2 \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \mathbf{x}_{n,m-1}^{j-1,P} \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{h^2}{M^2} \delta^2 \\
 &\quad + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \mathbf{x}_{n,m-1}^{j,P-1} \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j-1,P} - \mathbf{x}_{n,m-1}^{j-1,P-1} \right\|^2 \right].
 \end{aligned}$$

By setting $\eta = \frac{\alpha h}{M} = \frac{1}{10\kappa M}$, we have

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \mathbf{x}_{n,M}^{j,P} - \mathbf{x}_{n,M}^{j-1,P} \right\|^2 \right] \\
 &\leq \left(1 - \frac{\alpha h}{M} \right)^M \mathbb{E} \left[\left\| \mathbf{x}_{n,0}^{j,P} - \mathbf{x}_{n,0}^{j-1,P} \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{h^2}{M} \delta^2 \\
 &\quad + \sum_{m=1}^M \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \mathbf{x}_{n,m-1}^{j,P-1} \right\|^2 \right] \\
 &\quad + \sum_{m=1}^M \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j-1,P} - \mathbf{x}_{n,m-1}^{j-1,P-1} \right\|^2 \right] \\
 &\leq \exp(-\alpha h) \Delta_{n-1}^j + \left(4 + \frac{4}{\eta} \right) \frac{h^2}{M} \delta^2 + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M} \mathcal{E}_n^j + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M} \mathcal{E}_n^{j-1} \\
 &\leq (1 - 0.1\alpha h) \Delta_{n-1}^j + \left(4 + \frac{4}{\eta} \right) \frac{h^2}{M} \delta^2 + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M} \mathcal{E}_n^j + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M} \mathcal{E}_n^{j-1} \\
 &= \left(1 - \frac{0.01}{\kappa} \right) \Delta_{n-1}^j + 4 \left(\frac{1}{M} + 10\kappa \right) h^2 \delta^2 + 4 \left(\frac{1}{M} + 10\kappa \right) \beta^2 h^2 \mathcal{E}_n^j \\
 &\quad + 4 \left(\frac{1}{M} + 10\kappa \right) \beta^2 h^2 \mathcal{E}_n^{j-1}. \tag{13}
 \end{aligned}$$

In the following, we further decompose \mathcal{E}_n^j . For any $n = 0, \dots, N-1$, $j \in [J]$, $p = 2, \dots, P$, and $m = 1, \dots, M$, we can decompose $\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{j,p} - \mathbf{x}_{n,m}^{j,p-1} \right\|^2 \right]$ as follows. By definition (Line 12 or 18 in Algorithm 1), we have

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{j,p} - \mathbf{x}_{n,m}^{j,p-1} \right\|^2 \right] \\
 &= \frac{h^2}{M^2} \mathbb{E} \left[\left\| \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-1}) - \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-2}) \right\|^2 \right] \\
 &\leq \frac{h^2 m}{M^2} \sum_{m'=0}^{m-1} \mathbb{E} \left[\left\| \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-1}) - \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-2}) \right\|^2 \right] \\
 &\leq \frac{h^2 m}{M^2} \sum_{m'=0}^{m-1} 3 \left[\mathbb{E} \left[\left\| \nabla f(\mathbf{x}_{n,m'}^{j,p-1}) - \nabla f(\mathbf{x}_{n,m'}^{j,p-2}) \right\|^2 \right] + \mathbb{E} \left[\left\| \nabla f(\mathbf{x}_{n,m'}^{j,p-1}) - \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-1}) \right\|^2 \right] \right. \\
 &\quad \left. + \mathbb{E} \left[\left\| \nabla f(\mathbf{x}_{n,m'}^{j,p-2}) - \mathbf{s}(\mathbf{x}_{n,m'}^{j,p-2}) \right\|^2 \right] \right] \\
 &\leq 3\beta^2 h^2 \max_{m'=1,\dots,M} \mathbb{E} \left[\left\| \mathbf{x}_{n,m'}^{j,p-1} - \mathbf{x}_{n,m'}^{j,p-2} \right\|^2 \right] + 6\delta^2 h^2. \tag{14}
 \end{aligned}$$

Furthermore,

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,1} - \mathbf{x}_{n,m-1}^{j,0} \right\|^2 \right] \\
 &= \mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^j - \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^{j,0}) - \left(\mathbf{x}_{n-1,M}^{j-1} - \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^{j-1,P-1}) \right) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^j - \mathbf{x}_{n-1,M}^{j-1} \right\|^2 \right] + 2 \frac{h^2 m}{M^2} \sum_{m'=0}^{m-1} \mathbb{E} \left[\left\| \mathbf{s}(\mathbf{x}_{n,m'}^{j-1,P}) - \mathbf{s}(\mathbf{x}_{n,m'}^{j-1,P-1}) \right\|^2 \right] \\
 &\leq 2\Delta_{n-1}^j + 6\beta^2 h^2 \mathcal{E}_n^{j-1} + 12\delta^2 h^2.
 \end{aligned} \tag{15}$$

Combining (14) and (15), we have

$$\mathcal{E}_n^j = \mathbb{E} \left[\left\| \mathbf{x}_{n,m-1}^{j,P} - \mathbf{x}_{n,m-1}^{j,P-1} \right\|^2 \right] \leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^j + 6 \cdot 0.03^P \mathcal{E}_n^{j-1} + 6.6\delta^2 h^2. \tag{16}$$

Substitute it into (13), we have for any $j = 2, \dots, J, n = 1, \dots, N-1$,

$$\begin{aligned}
 \Delta_n^j &\leq \left(1 - \frac{0.01}{\kappa} + 8 \left(\frac{1}{M} + 10\kappa \right) 0.03^P \right) \Delta_{n-1}^j + 4.4 \left(\frac{1}{M} + 10\kappa \right) h^2 \delta^2 \\
 &\quad + 4.4 \left(\frac{1}{M} + 10\kappa \right) \beta^2 h^2 \mathcal{E}_n^{j-1}
 \end{aligned} \tag{17}$$

$$\leq \left(1 - \frac{0.005}{\kappa} \right) \Delta_{n-1}^j + 4.4 \left(\frac{1}{M} + 10\kappa \right) h^2 \delta^2 + 4.4 \left(\frac{1}{M} + 10\kappa \right) \beta^2 h^2 \mathcal{E}_n^{j-1}, \tag{18}$$

where the second inequality holds since $P \geq \frac{2 \log \kappa}{3} + 4$ implies $8 \left(\frac{1}{M} + 10\kappa \right) 0.03^P \leq \frac{0.005}{\kappa}$.

Decomposition when $j = 1$. When $j = 1$, similarly, we have for $p = 1, \dots, P$,

$$\mathbf{x}_{n,m+1}^{1,p} = \mathbf{x}_{n,m}^{1,p} - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m}^{1,p-1}) + \sqrt{2}(B_{nh+(m+1)/h} - B_{nh+mh/M}),$$

and

$$\mathbf{x}_{n,m+1}^0 = \mathbf{x}_{n,m}^0 - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n-1,M}^0) + \sqrt{2}(B_{nh+(m+1)/h} - B_{nh+mh/M}).$$

Thus by the contraction of $\phi(\mathbf{x}) = \mathbf{x} - \frac{h}{M} \nabla f(\mathbf{x})$ (Lemma 2.2 in Altschuler & Talwar (2023)), we have

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \mathbf{x}_{n,m+1}^{1,P} - \mathbf{x}_{n,m+1}^0 \right\|^2 \right] \\
 &= \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m}^{1,P-1}) - \left(\mathbf{x}_{n,m}^0 - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right) \right\|^2 \right] \\
 &\leq (1 + \eta) \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^{1,P}) - \left(\mathbf{x}_{n,m}^0 - \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^0) \right) \right\|^2 \right] \\
 &\quad + \left(2 + \frac{2}{\eta} \right) \mathbb{E} \left[\left\| \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^{1,P}) - \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^{1,P-1}) + \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^0) - \frac{h}{M} \nabla f(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] \\
 &\quad + \left(2 + \frac{2}{\eta} \right) \mathbb{E} \left[\left\| \frac{h}{M} \nabla f(\mathbf{x}_{n,m}^{1,P-1}) - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n,m}^{1,P-1}) + \frac{h}{M} \nabla f(\mathbf{x}_{n-1,M}^0) - \frac{h}{M} \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] \\
 &\leq (1 + \eta) \left(1 - \frac{\alpha h}{M} \right)^2 \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^0 \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{\delta^2 h^2}{M^2} \\
 &\quad + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^{1,P-1} \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2 h^2}{M^2} \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^0 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right].
 \end{aligned}$$

For third term $\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^{1,P-1} \right\|^2 \right]$, we have

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^{1,P-1} \right\|^2 \right] \\
 &= \mathbb{E} \left[\left\| \frac{h}{M} \sum_{m'=0}^m \mathbf{s}(\mathbf{x}_{n,m'}^{1,P-1}) - \mathbf{s}(\mathbf{x}_{n,m'}^{1,P-2}) \right\|^2 \right] \\
 &\leq \frac{mh^2}{M^2} \sum_{m'=0}^m \mathbb{E} \left[\left\| \mathbf{s}(\mathbf{x}_{n,m'}^{1,P-1}) - \mathbf{s}(\mathbf{x}_{n,m'}^{1,P-2}) \right\|^2 \right] \\
 &\leq 3\beta^2 h^2 \max_{m'=0,\dots,M} \mathbb{E} \left[\left\| \mathbf{x}_{n,m'}^{1,P-1} - \mathbf{x}_{n,m'}^{1,P-2} \right\|^2 \right] + 6\delta^2 h^2.
 \end{aligned}$$

Thus

$$\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^{1,P-1} \right\|^2 \right] \leq 0.03^{P-1} \max_{m'=0,\dots,M} \mathbb{E} \left[\left\| \mathbf{x}_{n,m'}^{1,1} - \mathbf{x}_{n,m'}^{1,0} \right\|^2 \right] + 6.2\delta^2 h^2. \quad (19)$$

For $\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,1} - \mathbf{x}_{n,m}^{1,0} \right\|^2 \right]$, by definition, we have

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,1} - \mathbf{x}_{n,m}^{1,0} \right\|^2 \right] \\
 &= \mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^1 - \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^0) - \left(\mathbf{x}_{n-1,M}^0 - \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^1 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right] + 2\mathbb{E} \left[\left\| \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n,m'}^0) - \frac{h}{M} \sum_{m'=0}^{m-1} \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^1 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right] + 2 \frac{h^2 m}{M^2} \sum_{m'=0}^{m-1} \mathbb{E} \left[\left\| \mathbf{s}(\mathbf{x}_{n,m'}^0) - \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] \\
 &\leq 2\mathbb{E} \left[\left\| \mathbf{x}_{n-1,M}^1 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right] + 6\beta^2 h^2 \max_{m' \in [M]} \mathbb{E} \left[\left\| \mathbf{x}_{n,m'}^0 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right] + 12\delta^2 h^2. \quad (20)
 \end{aligned}$$

For $\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^0 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right]$, by definition of $\mathbf{x}_{n,m}^0$ (Line 7 in Algorithm 1), we have

$$\begin{aligned}
 & \mathbb{E} \left[\left\| \mathbf{x}_{n,m}^0 - \mathbf{x}_{n-1,M}^0 \right\|^2 \right] \\
 &= \frac{h^2 m^2}{M^2} \mathbb{E} \left[\left\| \mathbf{s}(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] + \frac{dhm}{M} \\
 &\leq 2\delta^2 h^2 + 2h^2 \mathbb{E} \left[\left\| \nabla f(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] + dh \\
 &\leq 2\delta^2 h^2 + 2h^2 \left(2\beta d + \frac{4\beta^2}{\alpha} \text{KL}(\mu_{n-1,M}^0 \parallel \pi) \right) + dh \\
 &= 4h^2 \beta d + 2h^2 \delta^2 + \frac{8\beta^2 h^2}{\alpha} \text{KL}^0_{n-1} + dh, \quad (21)
 \end{aligned}$$

where the last inequality is implied from the following lemma, (Vempala & Wibisono, 2019, Lemma 10)

$$\mathbb{E} \left[\left\| \nabla f(\mathbf{x}_{n-1,M}^0) \right\|^2 \right] \leq 2\beta d + \frac{4\beta^2}{\alpha} \text{KL}(\mu_{n-1,M}^0 \parallel \pi).$$

Combining (19), (20), and (21), and $P \geq 4$, we have

$$\mathbb{E} \left[\left\| \mathbf{x}_{n,m}^{1,P} - \mathbf{x}_{n,m}^{1,P-1} \right\|^2 \right]$$

$$\begin{aligned}
 &\leq 0.03^{P-1} \max_{m'=0,\dots,M} \mathbb{E} \left[\left\| \mathbf{x}_{n,m'}^{1,1} - \mathbf{x}_{n,m'}^{1,0} \right\|^2 \right] + 6.2h^2\delta^2 \\
 &\leq 0.03^{P-1} \left[2\Delta_{n-1}^1 + 6\beta^2h^2 \left(4h^2\beta d + 2h^2\delta^2 + \frac{8\beta^2h^2}{\alpha} \text{KL}_{n-1}^0 + dh \right) + 12\delta^2h^2 \right] + 6.2h^2\delta^2 \\
 &\leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.01dh + 0.01 \frac{\beta^2h^2}{\alpha} \text{KL}_{n-1}^0.
 \end{aligned} \tag{22}$$

By setting $\eta = \frac{\alpha h}{M} = \frac{1}{10\kappa M}$, we have

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \mathbf{x}_{n,M}^{1,P} - \mathbf{x}_{n,M}^0 \right\|^2 \right] \\
 &\leq \left(1 - \frac{\alpha h}{M} \right)^M \mathbb{E} \left[\left\| \mathbf{x}_{n,0}^{1,P} - \mathbf{x}_{n,0}^0 \right\|^2 \right] + \left(4 + \frac{4}{\eta} \right) \frac{\delta^2h^2}{M} \\
 &\quad + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2h^2}{M} \left(2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.01dh + 0.01 \frac{\beta^2h^2}{\alpha} \text{KL}_{n-1}^0 \right) \\
 &\quad + \left(4 + \frac{4}{\eta} \right) \frac{\beta^2h^2}{M} \left(4h^2\beta d + 2h^2\delta^2 + \frac{8\beta^2h^2}{\alpha} \text{KL}_{n-1}^0 + dh \right) \\
 &\leq \left(1 - \frac{0.01}{\kappa} + 4 \left(\frac{1}{M} + 10\kappa \right) 0.03^P \right) \Delta_{n-1}^1 + \left(\frac{1}{M} + 10\kappa \right) \left(5\delta^2h^2 + 6\beta^2dh^3 + 0.4\beta^2h^2 \frac{\text{KL}_{n-1}^0}{\alpha} \right) \\
 &\leq \Delta_{n-1}^1 + \left(\frac{1}{M} + 10\kappa \right) \left(5\delta^2h^2 + 6\beta^2dh^3 + 0.4\beta^2h^2 \frac{\text{KL}_{n-1}^0}{\alpha} \right)
 \end{aligned}$$

where the last inequality holds since $P \geq \frac{2\log \kappa}{3} + 4$ implies $8 \left(\frac{1}{M} + 10\kappa \right) 0.03^P \leq \frac{0.005}{\kappa}$. \square

When $n = 0$, the update is identical to the Picard iteration shown in [Anari et al. \(2024\)](#), thus we have the following lemma.

Lemma B.6 (Lemma 18 in [Anari et al. \(2024\)](#)). *For $j = 1, \dots, J$, we have*

$$\Delta_0^j \leq 0.03^P \Delta_0^{j-1} + 6.2\delta^2h^2,$$

with $\Delta_0^0 := \max_{m=0,\dots,M} \mathbb{E} \left[\left\| \mathbf{x}_{0,m}^0 - \mathbf{x}_0 \right\|^2 \right] \leq \frac{4\beta^2h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 1.4dh + 2\delta^2h^2$.

Corollary B.7. *For $n = 1, \dots, N-1$, we have*

$$\Delta_n^1 \leq n \left(\frac{1}{M} + 10\kappa \right) \left(5.1\delta^2h^2 + 0.5 \frac{\beta^2h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 10\kappa^2\beta^2dh^3 \right).$$

Furthermore, for $j = 1, \dots, J$ and $n = 0$, we have

$$\Delta_0^j \leq 0.03^{jP} \frac{4\beta^2h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 1.4 \cdot 0.03^{jP} dh + 6.7\delta^2h^2.$$

Proof. By Lemma B.6, we have

$$\begin{aligned}
 \Delta_0^j &\leq 0.03^P \Delta_0^{j-1} + 6.2\delta^2h^2 \\
 &\leq 0.03^{jP} \Delta_0^0 + 6.6\delta^2h^2 \\
 &\leq 0.03^{jP} \left(\frac{4\beta^2h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 1.4dh + 2\delta^2h^2 \right) + 6.6\delta^2h^2 \\
 &\leq 0.03^{jP} \frac{4\beta^2h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 1.4 \cdot 0.03^{jP} dh + 6.7\delta^2h^2.
 \end{aligned}$$

Combining Lemma B.1 and Lemma B.5, we have

$$\begin{aligned}
 \Delta_n^1 &\leq \Delta_0^1 + \sum_{i=1}^n \left(\frac{1}{M} + 10\kappa \right) \left(5\delta^2 h^2 + 6\beta^2 dh^3 + 0.4\beta^2 h^2 \frac{\text{KL}_{i-1}^0}{\alpha} \right) \\
 &\leq \Delta_0^1 + n \left(\frac{1}{M} + 10\kappa \right) (5\delta^2 h^2 + 6\beta^2 dh^3) \\
 &\quad + \sum_{i=1}^n \left(\frac{1}{M} + 10\kappa \right) 0.4 \frac{\beta^2 h^2}{\alpha} \left(\exp(-\alpha n h) \text{KL}(\mu_0 \parallel \pi) + \frac{8\beta^2 dh}{\alpha} \right) \\
 &\leq \Delta_0^1 + n \left(\frac{1}{M} + 10\kappa \right) \left(5\delta^2 h^2 + 6\beta^2 dh^3 + 0.4 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 3.2\kappa^2 \beta^2 dh^3 \right) \\
 &\leq n \left(\frac{1}{M} + 10\kappa \right) \left(5.1\delta^2 h^2 + 0.5 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 10\kappa^2 \beta^2 dh^3 \right).
 \end{aligned}$$

□

B.3. One Step Analysis of \mathcal{E}_n^j

In this section, we analyze the one step change of \mathcal{E}_n^j .

Lemma B.8. For any $j = 2, \dots, J$, $n = 1, \dots, N-1$, we have

$$\mathcal{E}_n^j \leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^j + 2 \cdot 0.03^P \mathcal{E}_{n-1}^{j-1} + 7\delta^2 h^2.$$

Furthermore, for $n = 1, \dots, N-1$, we have

$$\mathcal{E}_n^1 \leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.01dh + 0.01 \frac{\beta^2 h^2}{\alpha} \text{KL}_{n-1}^0.$$

Proof. By (16), the first inequality holds. By (22), the second inequality holds. □

Corollary B.9. For $n = 1, \dots, N-1$, we have

$$\mathcal{E}_n^1 \leq n \left(5.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right).$$

Proof. Combining Lemma B.1, Lemma B.8 and Corollary B.7, we have

$$\begin{aligned}
 \mathcal{E}_n^1 &\leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.01dh + 0.01 \frac{\beta^2 h^2}{\alpha} \text{KL}_{n-1}^0 \\
 &\leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.01dh \\
 &\quad + 0.01 \frac{\beta^2 h^2}{\alpha} \left(\exp(-\alpha(n+1)h) \text{KL}(\mu_0 \parallel \pi) + \frac{8\beta^2 dh}{\alpha} \right) \\
 &\leq 2 \cdot 0.03^{P-1} \Delta_{n-1}^1 + 6.3h^2\delta^2 + 0.02\kappa dh + 0.01 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) \\
 &\leq 2 \cdot 0.03^{P-1} \left(n \left(\frac{1}{M} + 10\kappa \right) \left(5.1\delta^2 h^2 + 0.5 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 10\kappa^2 \beta^2 dh^3 \right) \right) \\
 &\quad + 6.3h^2\delta^2 + 0.02\kappa dh + 0.01 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) \\
 &\leq n \cdot 0.06 \left(5.1\delta^2 h^2 + 0.5 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right) \\
 &\quad + 6.3h^2\delta^2 + 0.02\kappa dh + 0.01 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) \\
 &\leq n \left(5.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right).
 \end{aligned}$$

where the fifth inequality holds since $P \geq \frac{2 \log \kappa}{3} + 4$ implies $\left(\frac{1}{M} + 10\kappa \right) 0.03^{P-1} \leq 0.03$. □

B.4. Proof of Theorem 4.2

We define an energy function as

$$L_n^j = \Delta_{n-1}^j + \kappa \mathcal{E}_n^{j-1}.$$

We note that $2 \cdot 0.03^{P-1} L_n^j + 7\delta^2 h^2 \geq \mathcal{E}_n^j$. By Lemma B.5 and Lemma B.8, we can decompose L_n^j as

$$\begin{aligned} L_n^j &= \Delta_{n-1}^j + \kappa \mathcal{E}_n^{j-1} \\ &\leq \left(1 - \frac{0.005}{\kappa}\right) \Delta_{n-2}^j + 4.4 \left(\frac{1}{M} + 10\kappa\right) h^2 \delta^2 + 4.4 \left(\frac{1}{M} + 10\kappa\right) \beta^2 h^2 \mathcal{E}_{n-1}^{j-1} \\ &\quad + \kappa(0.03^{P-1} \Delta_{n-1}^{j-1} + 2 \cdot 0.03^P \mathcal{E}_n^{j-2} + 7\delta^2 h^2) \\ &\leq \left(1 - \frac{0.005}{\kappa}\right) \Delta_{n-2}^j + \kappa \left(1 - \frac{0.005}{\kappa}\right) \mathcal{E}_{n-1}^{j-1} + \kappa \cdot 0.03^{P-1} \Delta_{n-1}^{j-1} + \kappa \cdot 0.03^{P-1} \cdot \kappa \mathcal{E}_n^{j-2} \\ &\quad + 56\kappa \delta^2 h^2 \\ &= \left(1 - \frac{0.005}{\kappa}\right) L_{n-1}^j + (\kappa \cdot 0.03^{P-1}) L_n^{j-1} + 56\kappa \delta^2 h^2. \end{aligned} \tag{23}$$

Combining $P \geq \frac{2 \log \kappa}{3} + 4$ implies $\kappa \cdot 0.03^{P-1} \leq 0.04$, we recursively bound L_n^j as

$$\begin{aligned} L_n^j &\leq \sum_{a=2}^n 0.04^{j-2} \binom{n-a+j-2}{j-2} L_a^2 + \sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} L_1^b \\ &\quad + \sum_{a=2}^j \sum_{b=2}^n \left(1 - \frac{0.001}{\kappa}\right)^{n-b} 0.04^{j-a} 65\kappa \delta^2 h^2 \\ &\leq \sum_{a=2}^n 0.04^{j-2} \binom{n-a+j-2}{j-2} L_a^2 + \sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} L_1^b \\ &\quad + 68000\kappa^2 \delta^2 h^2. \end{aligned} \tag{24}$$

For the first term $\sum_{a=2}^n 0.04^{j-2} \binom{n-a+j-2}{j-2} L_a^2$, we first bound L_a^2 . To do so, we first bound Δ_n^2 as follows. Combining Lemma B.5 and Corollary B.9, we have

$$\begin{aligned} \Delta_n^2 &\leq \left(1 - \frac{0.005}{\kappa}\right) \Delta_{n-1}^2 + 4.4 \left(\frac{1}{M} + 10\kappa\right) h^2 \delta^2 + 4.4 \left(\frac{1}{M} + 10\kappa\right) \beta^2 h^2 \mathcal{E}_n^1 \\ &\leq \Delta_{n-1}^2 + 48.4\kappa h^2 \delta^2 + 48.4\kappa \beta^2 h^2 \left(n \left(5.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right) \right) \\ &\leq \Delta_{n-1}^2 + 48.4\kappa \beta^2 h^2 n \left(55.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right) \\ &\leq \Delta_0^2 + 48.4\kappa \beta^2 h^2 n^2 \left(55.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right) \\ &\leq 0.03^{2P} \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 1.4 \cdot 0.03^{2P} dh + 6.7\delta^2 h^2 \\ &\quad + 48.4\kappa \beta^2 h^2 n^2 \left(55.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.1\kappa^2 dh \right) \\ &\leq 48.4\kappa \beta^2 h^2 n^2 \left(67.2\delta^2 h^2 + 0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \parallel \pi) + 0.2\kappa^2 dh \right). \end{aligned}$$

Thus

$$\begin{aligned}
 L_a^2 &= \Delta_{a-1}^2 + \kappa \mathcal{E}_a^1 \\
 &\leq 0.49\kappa(a-1)^2 \left(67.2\delta^2 h^2 + 0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.2\kappa^2 dh \right) \\
 &\quad + \kappa \left(a \left(5.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.1\kappa^2 dh \right) \right) \\
 &\leq \kappa a^2 \left(39\delta^2 h^2 + 0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.2\kappa^2 dh \right).
 \end{aligned}$$

Thus by $\binom{m}{n} \leq \left(\frac{em}{n}\right)^n$ for $m \geq n > 0$, we have

$$\begin{aligned}
 &\sum_{a=2}^n 0.04^{j-2} \binom{n-a+j-2}{j-2} L_a^2 \\
 &\leq \sum_{a=2}^n 0.04^{j-2} e^{j-2} \left(\frac{n-a+j-2}{j-2} \right)^{j-2} L_a^2 \\
 &\leq \sum_{a=2}^n 0.04^{j-2} e^{2j-4} L_a^2 \\
 &\leq \sum_{a=2}^n 0.3^{j-2} \kappa a^2 \left(39\delta^2 h^2 + 0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.2\kappa^2 dh \right) \\
 &\leq 0.3^{j-2} \kappa n^3 \left(39\delta^2 h^2 + 0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.2\kappa^2 dh \right). \tag{25}
 \end{aligned}$$

For the second term $\sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} L_1^b$, we first bound L_1^b . Firstly, for \mathcal{E}_1^{b-1} , combining Corollary B.7 and Corollary B.9, we have

$$\begin{aligned}
 \mathcal{E}_1^{b-1} &\leq 2 \cdot 0.03^{P-1} \Delta_0^{b-1} + 2 \cdot 0.03^P \mathcal{E}_1^{b-2} + 7\delta^2 h^2 \\
 &\leq 2 \cdot 0.03^{P-1} \left(0.03^{(b-1)P} \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 1.4 \cdot 0.03^{(b-1)P} dh + 6.7\delta^2 h^2 \right) \\
 &\quad + 2 \cdot 0.03^P \mathcal{E}_1^{b-2} + 7\delta^2 h^2 \\
 &\leq 2 \cdot 0.03^P \mathcal{E}_1^{b-2} + 0.03^b \left(0.01 \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.01 dh \right) + 7.1\delta^2 h^2 \\
 &\leq (2 \cdot 0.03^P)^{b-2} \mathcal{E}_1^1 + \sum_{i=0}^{b-3} (2 \cdot 0.03^P)^i \left(0.03^{b-i} \left(0.01 \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.01 dh \right) + 7.1\delta^2 h^2 \right) \\
 &\leq (2 \cdot 0.03^P)^{b-2} \mathcal{E}_1^1 + \sum_{i=0}^{b-3} 0.01^i 0.03^i \left(0.03^{b-i} \left(0.01 \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.01 dh \right) + 7.1\delta^2 h^2 \right) \\
 &\leq (2 \cdot 0.03^P)^{b-2} \left(5.5\delta^2 h^2 + 0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.1\kappa^2 dh \right) \\
 &\quad + 0.03^b \left(0.02 \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.02 dh \right) + 7.2\delta^2 h^2 \\
 &\leq 0.03^b \left(0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 0.1 dh \right) + 7.3\delta^2 h^2.
 \end{aligned}$$

As for Δ_0^b we have

$$\Delta_0^b \leq 0.03^{bP} \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \|\pi) + 1.4 \cdot 0.03^{bP} dh + 6.7\delta^2 h^2.$$

Thus, we bound the first term as

$$\begin{aligned}
 L_1^b &= \Delta_0^b + \kappa \mathcal{E}_1^{b-1} \\
 &\leq 0.03^{bP} \frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 1.4 \cdot 0.03^{bP} dh + 6.7\delta^2 h^2 \\
 &\quad + \kappa 0.03^b \left(0.1 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.1 dh \right) + 7.3\delta^2 h^2 \\
 &\leq \kappa 0.03^b \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) + 14\delta^2 h^2.
 \end{aligned}$$

Thus by $\binom{m}{n} \leq \left(\frac{em}{n}\right)^n$ for $m \geq n > 0$, and $\sum_{i=0}^m \binom{n+i}{n} x^i = \frac{1-(m+1)x + \binom{m+n+1}{n} B_x(m+1, n+1)}{(1-x)^{n+1}} \leq \frac{1}{(1-x)^{n+1}}$ we have

$$\begin{aligned}
 &\sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} L_1^b \\
 &\leq \sum_{b=2}^j 0.04^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} \left(\kappa 0.03^b \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) \right) \\
 &\quad + \sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} 14\delta^2 h^2 \\
 &\leq \sum_{b=2}^j 0.04^j \binom{n-1+j-b}{j-b} \kappa \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) \\
 &\quad + \sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} 14\delta^2 h^2 \\
 &\leq \sum_{i=0}^{j-2} 0.04^j e^i \left(1 + \frac{n-1}{i}\right)^i \kappa \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) \\
 &\quad + \sum_{i=0}^{j-2} (\kappa \cdot 0.03^{P-1})^i \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+i}{i} 14\delta^2 h^2 \\
 &\leq 0.11^j e^{n-1} \kappa \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) \\
 &\quad + \frac{1}{(1 - \kappa \cdot 0.03^{P-1})^n} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} (6.6 + 7.9\kappa) \delta^2 h^2 \\
 &\leq 0.11^j e^{n-1} \kappa \left(0.2 \frac{\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 0.2 dh \right) + \frac{1}{(1 - \kappa \cdot 0.03^{P-1})} (6.6 + 7.9\kappa) \delta^2 h^2 \\
 &\leq 0.11^j e^{n-1} \left(2.2\kappa \left(\frac{4\beta^2 h^2}{\alpha} \text{KL}(\mu_0 \| \pi) + 1.6 dh + 2\delta^2 h^2 \right) \right) + 20\kappa \delta^2 h^2,
 \end{aligned}$$

where the second-to-last inequality is implied by $8 \left(\frac{1}{M} + 10\kappa \right) 0.03^P \leq \frac{0.005}{\kappa}$.

Combing (24) and (25), we bound L_n^j as

$$\begin{aligned}
 L_n^j &\leq \sum_{a=2}^n 0.04^{j-2} \binom{n-a+j-2}{j-2} L_a^2 + \sum_{b=2}^j (\kappa \cdot 0.03^{P-1})^{j-b} \left(1 - \frac{0.005}{\kappa}\right)^{n-1} \binom{n-1+j-b}{j-b} L_1^b \\
 &\quad + 68000\kappa^2\delta^2h^2 \\
 &\leq 0.3^{j-2}\kappa n^3 \left(39\delta^2h^2 + 0.2\frac{\beta^2h^2}{\alpha}\text{KL}(\mu_0\|\pi) + 0.2\kappa^2dh\right) \\
 &\quad + 0.11^je^{n-1} \left(2.2\kappa \left(\frac{4\beta^2h^2}{\alpha}\text{KL}(\mu_0\|\pi) + 1.6dh + 2\delta^2h^2\right)\right) + 20\kappa\delta^2h^2 + 68000\kappa^2\delta^2h^2 \\
 &\leq 0.3^{j-2}e^{n-1}\kappa n^3 (41\delta^2h^2 + 1.8\kappa^2dh + 0.5\kappa h\text{KL}(\mu_0\|\pi)) + 68020\kappa^2\delta^2h^2.
 \end{aligned}$$

Since $8\left(\frac{1}{M} + 10\kappa\right)0.03^P \leq \frac{0.005}{\kappa}$ implies $\kappa^20.03^{P-1} \leq 0.003$, we have

$$\begin{aligned}
 \mathcal{E}_n^j &\leq 2 \cdot 0.03^{P-1} L_n^j + 7\delta^2h^2 \\
 &\leq 2 \cdot 0.03^{P-1} (0.3^{j-2}e^{n-1}\kappa n^3 (41\delta^2h^2 + 1.8\kappa^2dh + 0.5\kappa h\text{KL}(\mu_0\|\pi)) + 68020\kappa^2\delta^2h^2) + 7\delta^2h^2 \\
 &\leq 0.3^{j-2}e^{n-1}n^3 (\delta^2h^2 + h\text{KL}(\mu_0\|\pi) + \kappa dh) + 416\delta^2h^2.
 \end{aligned}$$

Thus when $J - N \geq \log\left(N^3 \left(\frac{\kappa\delta^2h + \kappa\text{KL}(\mu_0\|\pi) + \kappa^2d}{\varepsilon^2}\right)\right)$, we have for any $n = 0, \dots, N-1$

$$\mathcal{E}_n^J \leq \frac{\varepsilon^2}{5\kappa\beta} + 416\delta^2h^2.$$

Recall

$$\text{KL}_{N-1}^J \leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2h\Delta_0^J) + 5\kappa\beta\mathcal{E} + \frac{0.6\beta d}{\alpha M} + \frac{28\delta^2}{\alpha},$$

thus when $\delta^2 \leq \frac{\alpha\varepsilon^2}{29}$, $M \geq \frac{\kappa d}{\varepsilon^2}$, and $N \geq 10\kappa \log \frac{\text{KL}(\mu_0\|\pi)}{\varepsilon^2}$, we have

$$\begin{aligned}
 \text{KL}_{N-1}^J &\leq e^{-1.2\alpha(N-1)h} (\text{KL}(\mu_0\|\pi) + 4.4\beta^2h\Delta_0^J) + 5\kappa\beta\mathcal{E} + \frac{0.6\beta d}{\alpha M} + \frac{28\delta^2}{\alpha} \\
 &\leq e^{-1.2\alpha(N-1)h} \left(\text{KL}(\mu_0\|\pi) + 4.4\beta^2h \left(0.03^{JP} \frac{4\beta^2h^2}{\alpha} \text{KL}(\mu_0\|\pi) + 1.4 \cdot 0.03^{JP} dh + 6.7\delta^2h^2 \right) \right) \\
 &\quad + 5\kappa\beta\mathcal{E} + \frac{0.6\beta d}{\alpha M} + \frac{28\delta^2}{\alpha} \\
 &\leq e^{-1.2\alpha(N-1)h} \text{KL}(\mu_0\|\pi) + \varepsilon^2 + 5\kappa\beta\mathcal{E} + \frac{0.6\beta d}{\alpha M} + \frac{29\delta^2}{\alpha} \\
 &\leq 5\varepsilon^2.
 \end{aligned}$$

C. Missing Details for Sampling for Diffusion Models

In this section, we begin by presenting the algorithm details in Appendix C.1. In Appendix C.2, following the approach of Chen et al. (2024), we apply Girsanov's Theorem and the interpolation method to decompose the KL divergence and bound the discretization error, accounting for the influence of the step size scheme and the estimation error of the score function. Finally, in Appendix C.3, we analyze the additional parallelization error and derive the overall error bound.

C.1. Algorithm

In the parallel Picard method for diffusion model, we use the similar parallelization across time slices as illustrated in Figure 1. In Lines 2–6, we generate the noises and fix them. In Lines 7–10, we initialize the value at the grid via sequential method with a stepsize $h_n = \mathcal{O}(1)$. In Lines 12–21, we update the grids diagonally, using the exponential integrator in Lines 14 and 19 instead of the Euler-Maruyama scheme. The step size scheme also differs from that used for log-concave

sampling. Here, we follow the discretization scheme with early stopping and exponential decay described in [Chen et al. \(2024, Section 3.1.2\)](#).

Algorithm 2 Parallel Picard Iteration Method for diffusion models

```

1: Input:  $\hat{\mathbf{y}}_0 \sim \widehat{\mathbf{q}}_0 = \mathcal{N}(0, I_d)$ , the learned NN-based score function  $\mathbf{s}_t^\theta(\cdot)$ , the depth of Picard iterations  $J$ , the depth of
   inner Picard iteration  $P$ , and a discretization scheme  $(T, (h_n)_{n=1}^N$  and  $(\tau_{n,m})_{n \in [0:N-1], m \in [0:M]}$ ).
2: for  $n = 0, \dots, N - 1$  do
3:   for  $m = 0, \dots, M$  (in parallel) do
4:      $\boldsymbol{\xi}_{n,m} \sim \mathcal{N}(0, I_d)$ 
5:   end for
6: end for
7: for  $n = 0, \dots, N - 1$  do
8:   for  $m = 0, \dots, M_n$  (in parallel) do
9:      $\hat{\mathbf{y}}_{-1,M}^j = \hat{\mathbf{y}}_0$ , for  $j = 0, \dots, J$ ,
       
$$\begin{aligned} \hat{\mathbf{y}}_{n,\tau_{n,m}}^0 &= e^{\frac{\tau_{n,m}}{2}} \hat{\mathbf{y}}_{n-1,\tau_{n,M}}^0 \\ &+ \sum_{m'=0}^{m-1} e^{\frac{\tau_{n,m}-\tau_{n,m'}+1}{2}} \left[ 2(e^{\epsilon_{n,m'}} - 1) \mathbf{s}_{t_n+\tau_{n,m'}}^\theta(\hat{\mathbf{y}}_{n-1,\tau_{n,M}}^0) + \sqrt{e^{\epsilon_{n,m'}} - 1} \boldsymbol{\xi}_{m'} \right], \end{aligned} \quad (26)$$

10:   end for
11: end for
12: for  $k = 1, \dots, N$  do
13:   for  $j = 1, \dots, \min\{k-1, J\}$  and  $m = 0, \dots, M_n$  (in parallel) do
14:     let  $n = k - j$ , and  $\hat{\mathbf{y}}_{n,0}^j = \hat{\mathbf{y}}_{n-1,M_n}^j$ ,
       
$$\begin{aligned} \hat{\mathbf{y}}_{n,\tau_{n,m}}^j &= e^{\frac{\tau_{n,m}}{2}} \hat{\mathbf{y}}_{n,0}^j \\ &+ \sum_{m'=0}^{m-1} e^{\frac{\tau_{n,m}-\tau_{n,m'}+1}{2}} \left[ 2(e^{\epsilon_{n,m'}} - 1) \mathbf{s}_{t_n+\tau_{n,m'}}^\theta(\hat{\mathbf{y}}_{n,\tau_{n,m'}}^{j-1}) + \sqrt{e^{\epsilon_{n,m'}} - 1} \boldsymbol{\xi}_{m'} \right], \end{aligned} \quad (27)$$

15:   end for
16: end for
17: for  $k = N + 1, \dots, N + J - 1$  do
18:   for  $n = \max\{0, k - J\}, \dots, N - 1$  and  $m = 0, \dots, M_n$  (in parallel) do
19:     let  $j = k - n$ , and  $\hat{\mathbf{y}}_{n,0}^j = \hat{\mathbf{y}}_{n-1,M_n}^j$ ,
       
$$\begin{aligned} \hat{\mathbf{y}}_{n,\tau_{n,m}}^j &= e^{\frac{\tau_{n,m}}{2}} \hat{\mathbf{y}}_{n,0}^j \\ &+ \sum_{m'=0}^{m-1} e^{\frac{\tau_{n,m}-\tau_{n,m'}+1}{2}} \left[ 2(e^{\epsilon_{n,m'}} - 1) \mathbf{s}_{t_n+\tau_{n,m'}}^\theta(\hat{\mathbf{y}}_{n,\tau_{n,m'}}^{j-1}) + \sqrt{e^{\epsilon_{n,m'}} - 1} \boldsymbol{\xi}_{m'} \right], \end{aligned} \quad (28)$$

20:   end for
21: end for
22: Return:  $\hat{\mathbf{y}}_{N-1,M_{N-1}}^J$ .
    
```

Stepsize scheme. We first present the stepsize schedule for diffusion models, which is the same as the discretization scheme in [Chen et al. \(2024\)](#). Specifically, we split the the time horizon T into N time slices with length $h_n \leq h = \frac{T}{N} = \Omega(1)$, and a large gap grid $(t_n)_{n=0}^N$ with $t_n = \sum_{i=1}^n h_i$. For any $n \in [0 : N - 1]$, we further split the n -th time slice into a grid $(\tau_{n,m})_{m=0}^{M_n}$ with $\tau_{n,0} = 0$ and $\tau_{n,M_n} = h_n$. We denote the step size of the m -th step in the n -th time slice as $\epsilon_{n,m} = \tau_{n,m+1} - \tau_{n,m}$, and let the total number of grids in the n -th time slice as M_n . The grids $(\tau_{n,m})_{m=0}^{M_n}$ is scheduled as follows,

1. for the first $N - 1$ time slice, we use the uniform discretization: for $n = 0, \dots, N - 2$ and $m = 0, \dots, M - 1$.

$$h_n = h, \quad \epsilon_{n,m} = \epsilon, \text{ and } M_n = M = \frac{h}{\epsilon},$$

- for the last time slice, we apply early stopping and exponential decay:

$$h_{N-1} = h - \delta, \quad \epsilon_{N-1,m} \leq \epsilon \wedge \epsilon(h - \tau_{N-1,m+1}).$$

We also define the indexing functions as follows: for $\tau \in [t_n, t_{n+1}]$, we define $I_n(\tau) \in \mathbb{N}$ such that $\sum_{j=1}^{I_n(\tau)} \epsilon_{n,j} \leq \tau < \sum_{j=1}^{I_n(\tau)+1} \epsilon_{n,j}$. We further define a piecewise function g such that $g_n(\tau) = \sum_{j=1}^{I_n(\tau)} \epsilon_{n,j}$ and thus we have $I_n(\tau) = \lfloor \tau/\epsilon \rfloor$ and $g_n(\tau) = \lfloor \tau/\epsilon \rfloor \epsilon$.

Exponential integrator for Picard iterations. Compared with Line 12 and Line 18 in Algorithm 1, where we use a forward Euler-Maruyama scheme for Picard iterations, we use the following exponential integrator scheme (Zhang & Chen, 2023; Chen et al., 2024). Specifically, In n -th time slice $[t_n, t_n + \tau_{n,M_n}]$, for each grid $t_n + \tau_{n,m}$, we simulate the approximated backward process (3) with Picard iterations as

$$\begin{aligned} \hat{\mathbf{y}}_{n,\tau_{n,m}}^{j+1} &= e^{\frac{\tau_{n,m}}{2}} \hat{\mathbf{y}}_{n-1,\tau_{n,M}}^{j+1} \\ &+ \sum_{m'=0}^{m-1} e^{\frac{\tau_{n,m}-\tau_{n,m'+1}}{2}} \left[2(e^{\epsilon_{n,m'}} - 1) \mathbf{s}_{t_n+\tau_{n,m'}}^\theta(\hat{\mathbf{y}}_{n-1,\tau_{n,M}}^j) + \sqrt{e^{\epsilon_{n,m'}} - 1} \boldsymbol{\xi}_{m'} \right]. \end{aligned}$$

We note such update also inherently allows for parallelization for $m = 1, \dots, M_n$.

C.2. Interpolation Processes and Decomposition of KL Divergence

Following the proof flow in Chen et al. (2024), we define the following processes:

- the original backward process,

$$d\tilde{\mathbf{x}}_t = \left[\frac{1}{2} \tilde{\mathbf{x}}_t + \nabla \log \tilde{p}_t(\tilde{\mathbf{x}}_t) d\mathbf{t} \right] + d\mathbf{w}_t, \quad \text{with } \tilde{\mathbf{x}}_0 \sim p_T; \quad (29)$$

- the approximated backward process,

$$d\mathbf{y}_t = \left[\frac{1}{2} \mathbf{y}_t + \mathbf{s}_t^\theta(\mathbf{y}_t) \right] dt + d\mathbf{w}_t, \quad \text{with } \mathbf{y}_0 \sim \mathcal{N}(0, I_d);$$

- the interpolation processes $(\hat{\mathbf{y}}_{t_n,\tau}^j)_{\tau \in [0,h]}$ over $\tau \in [0, h]$ conditioned on the filtration of the backward SDE (29) up to time t \mathcal{F}_t , for any fixed $n = 0, \dots, N-1, j = 1, \dots, J$,

$$d\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) + \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-1}(\omega)) \right] d\tau + d\mathbf{w}_{t_n+\tau}(\omega); \quad (30)$$

$$\text{with } \hat{\mathbf{y}}_{t_n,0}^j(\omega) = \hat{\mathbf{y}}_{t_{n-1},\tau_{n-1},M_{n-1}}^j(\omega).$$

- the initialization process,

$$d\hat{\mathbf{y}}_{t_n,\tau}^0(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n,\tau}^0(\omega) + \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_{n-1},\tau_{n-1},M}^0(\omega)) \right] d\tau + d\mathbf{w}_{t_n+\tau}(\omega), \quad (31)$$

$$\text{with } \hat{\mathbf{y}}_{t_0,0}^0 = \hat{\mathbf{y}}_0 \text{ and } \hat{\mathbf{y}}_{t_n,0}^0 = \hat{\mathbf{y}}_{t_{n-1},\tau_{n-1},M}.$$

Remark C.1. The main difference compared to the auxiliary process defined in Chen et al. (2024) is the change of the start point across each update.

We can demonstrate that the interpolation processes remain well-defined after parallelization across time slices.

Lemma C.2. *The auxiliary process $(\hat{\mathbf{y}}_{t_n, \tau}^j(\omega))_{\tau \in [0, h_n]}$ is $\mathcal{F}_{t_n + \tau}$ -adapted for any $j = 1, \dots, J$ and $n = 0, \dots, n-1$.*

Proof. Since the initialization $\hat{\mathbf{y}}_{t_n, \tau}^0(\omega)$ satisfies

$$d\hat{\mathbf{y}}_{t_n, \tau}^0(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n, \tau}^0(\omega) + \mathbf{s}_{t_n + g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_{n-1}, \tau_{n-1}, M}^0(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n + \tau}(\omega),$$

we can claim $\hat{\mathbf{y}}_{t_n, \tau}^0(\omega)$ is $\mathcal{F}_{t_n + \tau}$ -adapted. Assume $\mathbf{y}_{t_n, \tau}$ is $\mathcal{F}_{t_n + \tau}$ -adapted, by $g_n(\tau) \leq \tau$, the Itô integral $\int_0^\tau \mathbf{s}_{t_n + g_n(\tau')}^\theta(\mathbf{y}_{t_n, g_n(\tau')}) d\tau'$ is well-defined and $\mathcal{F}_{t_n + \tau}$ -adapted. Therefore SDE

$$d\mathbf{y}'_{t_n, \tau}(\omega) = \left[\frac{1}{2} \mathbf{y}'_{t_n, \tau}(\omega) + \mathbf{s}_{t_n + g_n(\tau)}^\theta \left(\mathbf{y}_{t_n, g_n(\tau)}(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n + \tau}(\omega)$$

has a unique strong solution $(\mathbf{y}'_{t_n, \tau}(\omega))_{\tau \in [0, h_n]}$ that is also $\mathcal{F}_{t_n + \tau}$ -adapted. The lemma is established through induction. \square

Finally, the following lemma shows the equivalence of our update rule and the auxiliary process, *i.e.*, the auxiliary process is an interpolation of the discrete points.

Lemma C.3. *For any $n = 0, \dots, N-1$, the update rule ((26)) in Algorithm 2 and the update rule ((27) or (28)) are equivalent to the exact solution of the auxiliary process (31), and (30) respectively, for any $j = 1, \dots, J$, and $\tau \in [0, h_n]$.*

Proof. Due to the similarity, we only prove the equivalence of the update rule ((26)). The dependency on ω will be omitted in the proof below.

For SDE (30), by multiplying $e^{-\frac{\tau}{2}}$ on both sides then integrating on both side from 0 to τ , we have

$$e^{-\frac{\tau}{2}} \hat{\mathbf{y}}_{t_n, \tau}^j - \hat{\mathbf{y}}_{t_n, 0}^j = \sum_{m=0}^{M_n} 2 \left(e^{-\frac{\tau \wedge \tau_{n, m}}{2}} - e^{-\frac{\tau \wedge \tau_{n, m+1}}{2}} \right) \mathbf{s}_{t_n + \tau_{n, m}}^\theta \left(\hat{\mathbf{y}}_{t_n, \tau_{n, m}}^{j-1} \right) + \int_0^\tau e^{-\frac{\tau'}{2}} d\mathbf{w}_{t_n + \tau'}.$$

Thus

$$\begin{aligned} \hat{\mathbf{y}}_{t_n, \tau}^j &= e^{\frac{\tau}{2}} \hat{\mathbf{y}}_{t_n, 0}^j + \sum_{m=0}^{M_n} 2 \left(e^{-\frac{\tau \wedge \tau_{n, m} - \tau \wedge \tau_{n, m+1}}{2}} - 1 \right) e^{\frac{0 \vee (\tau - \tau_{n, m+1})}{2}} \mathbf{s}_{t_n + \tau_{n, m}}^\theta \left(\hat{\mathbf{y}}_{t_n, \tau_{n, m}}^{j-1} \right) \\ &\quad + \sum_{m=0}^{M_n} \int_{\tau \wedge \tau_{n, m}}^{\tau \wedge \tau_{n, m+1}} e^{\frac{\tau - \tau'}{2}} d\mathbf{w}_{t_n + \tau'}. \end{aligned}$$

By Itô isometry and let $\tau = \tau_{n, m}$ we get the desired result. \square

C.2.1. DECOMPOSITION OF KL DIVERGENCE

Similar as the analysis in Section B.2 of [Chen et al. \(2024\)](#), we conclude the following lemma by Corollary A.3.

Lemma C.4. *Assume $\delta_{t_n}(\tau, \omega) = \mathbf{s}_{t_n + g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^{J-1}(\omega)) - \nabla \log \tilde{p}_{t_n + \tau}(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega))$. Then we have the following one-step decomposition,*

$$\text{KL}(\tilde{p}_{t_{n+1}} \parallel \hat{q}_{t_{n+1}}) \leq \text{KL}(\tilde{p}_{t_n} \parallel \hat{q}_{t_n}) + \mathbb{E}_{\omega \sim q | \mathcal{F}_{t_n}} \left[\frac{1}{2} \int_0^{h_n} \|\delta_{t_n}(\tau, \omega)\|^2 d\tau \right].$$

Now, the problem remaining is reduced to bound the following discrepancy,

$$\begin{aligned}
 & \int_0^{h_n} \|\delta_{t_n}(\tau, \omega)\|^2 d\tau \\
 &= \int_0^{h_n} \left\| \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^{J-1}(\omega)) - \nabla \log \tilde{p}_{t_n+\tau}(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) \right\|^2 d\tau \\
 &\leq 3 \left(\underbrace{\int_0^{h_n} \left\| \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^J(\omega)) - \nabla \log \tilde{p}_{t_n+\tau}(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) \right\|^2 d\tau}_{:=A_{t_n}(\omega)} \right. \\
 &\quad + \underbrace{\int_0^{h_n} \left\| \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^J(\omega)) - \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^J(\omega)) \right\|^2 d\tau}_{:=B_{t_n}(\omega)} \\
 &\quad \left. + \underbrace{\int_0^{h_n} \left\| \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^J(\omega)) - \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^{J-1}(\omega)) \right\|^2 d\tau}_{:=C_{t_n}(\omega)} \right), \tag{32}
 \end{aligned}$$

where $A_{t_n}(\omega)$ measures the discretization error, $B_{t_n}(\omega)$ measures the estimation error of score function, and $C_{t_n}(\omega)$ measures the error by Picard iteration.

C.2.2. DISCRETIZATION ERROR AND ESTIMATION ERROR OF SCORE FUNCTION IN EVERY TIME SLICE

The following lemma from Benton et al. (2024); Chen et al. (2024) bounds the expectation of the discretization error A_{t_n} .

Lemma C.5 (Discretization error (Benton et al., 2024, Section 3.1) or (Chen et al., 2024, Lemma B.7)). *We have for $n \in [0 : N - 2]$*

$$\mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} [A_{t_n}(\omega)] \lesssim \epsilon d h_n,$$

and

$$\mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} [A_{t_{N-1}}(\omega)] \lesssim \epsilon d \log \eta^{-1},$$

where η is the parameter for early stopping.

The following lemma from Chen et al. (2024) bounds the expectation of the estimation error of score function, B_{t_n} , and we restate the proof for the convenience.

Lemma C.6 (Estimation error of score function (Chen et al., 2024, Section B.3)). $\sum_{n=0}^{N-1} \mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} [B_{t_n}] \leq \delta_2^2$.

Proof. By Assumption 5.1 and the fact that the process $\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)$ follows the backward SDE with the true score function under the measure \tilde{p} , we have

$$\begin{aligned}
 & \sum_{n=1}^{N-1} \mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} [B_{t_n}(\omega)] \\
 &\leq \mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} \left[\sum_{n=1}^{N-1} \int_0^{h_n} \left\| \mathbf{s}_{t_n+\tau}^\theta(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) - \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) \right\|^2 d\tau \right] \\
 &= \mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} \left[\sum_{n=1}^{N-1} \sum_{m=0}^{M_n} \epsilon_{n,m} \left\| \mathbf{s}_{t_n+\tau}^\theta(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) - \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n, \tau}^J(\omega)) \right\|^2 d\tau \right] \\
 &= \mathbb{E}_{\omega \sim \tilde{p}|\mathcal{F}_{t_n}} \left[\sum_{n=0}^{N-1} \sum_{m=0}^{M_n} \epsilon_{n,m} \left\| \mathbf{s}_{t_n+\tau}^\theta(\tilde{\mathbf{x}}_{t_n+\tau}(\omega)) - \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\tilde{\mathbf{x}}_{t_n+\tau}(\omega)) \right\|^2 d\tau \right] \\
 &\leq \delta_2^2.
 \end{aligned}$$

□

C.2.3. ANALYSIS FOR INITIALIZATION

By setting the depth of iteration as $J = 1$ in [Chen et al. \(2024\)](#), our initialization parts (Lines 4-7 in Algorithm 2) and the initialization process ((31)) are identical to the Algorithm 1 and the auxiliary process (Definition B.1) in [Chen et al. \(2024\)](#). We provide a brief overview of their analysis and reformulate it to align with our initialization. Let

$$A_{t_n}^0(\omega) := \int_0^{h_n} \left\| \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^0(\omega)) - \nabla \log \tilde{p}_{t_n+\tau}(\hat{\mathbf{y}}_{t_n,\tau}^0(\omega)) \right\|^2 d\tau$$

and

$$B_{t_n}^0(\omega) := \int_0^{h_n} \left\| \mathbf{s}_{t_n+g_n(\tau)}^\theta(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^0(\omega)) - \nabla \log \tilde{p}_{t_n+g_n(\tau)}(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^0(\omega)) \right\|^2 d\tau$$

Lemma C.7 (Lemma B.5 or Lemma B.6 with $K = 1$ in [Chen et al. \(2024\)](#)). *For any $n = 0, \dots, N-1$, suppose the initialization $\hat{\mathbf{y}}_{t_n,0}^0$ follows the distribution of $\tilde{x}_{t_n} \sim \tilde{p}_{t_n}$, if $3e^{\frac{7}{2}h_n}h_nL_s < 0.5$, then the following estimate*

$$\begin{aligned} \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{t_n,\tau}^0(\omega) - \hat{\mathbf{y}}_{t_n,0}^0(\omega) \right\|^2 \right] &\leq 2h_n e^{\frac{7}{2}h_n} (M_s + 2d) \\ &\quad + 6e^{\frac{7}{2}h_n} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} [A_{t_n}^0(\omega) + B_{t_n}^0(\omega)]. \end{aligned}$$

Furthermore, the $A_{t_n}^0(\omega)$ and $B_{t_n}^0(\omega)$ can be bounded as

Lemma C.8 (([Chen et al., 2024](#), Lemma B.7)). *We have for $n \in [0 : N-2]$*

$$\mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} [A_{t_n}^0(\omega)] \lesssim \epsilon d h_n,$$

and

$$\mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} [A_{t_{N-1}}^0(\omega)] \lesssim \epsilon d \log \eta^{-1},$$

where η is the parameter for early stopping.

Lemma C.9 (([Chen et al., 2024](#), Section B.3)). $\sum_{n=1}^{N-1} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} [B_{t_n}^0(\omega)] \leq \delta_2^2$.

Thus we have the following uniform bound for our initialization.

Corollary C.10. *With the same assumption in Lemma C.7, we have*

$$\sup_{n=0,\dots,N} \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{t_n,\tau}^0(\omega) - \hat{\mathbf{y}}_{t_n,0}^0(\omega) \right\|^2 \right] \lesssim d.$$

C.3. Convergence of Picard iteration

Similarly, we define

$$\mathcal{E}_n^j = \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2 \right],$$

and

$$\Delta_n^j = \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{t_n,\tau_{n,M}}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau_{n,M}}^{j-1}(\omega) \right\|^2 \right].$$

Furthermore, we let $\mathcal{E}_I = \sup_{n=0,\dots,N-1} \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n,\tau}^0 - \hat{\mathbf{y}}_{n-1,\tau_{n,M}}^0 \right\|^2 \right]$. We note that by Corollary C.10, $\mathcal{E}_I \lesssim d$.

Lemma C.11 (One-step decomposition of \mathcal{E}_n^j). *Assume $L_s^2 e^{2h_n} h_n \leq 0.01$ and $e^{2h_n} \leq 2$. For any $j = 2, \dots, J$, $n = 0, \dots, N-1$, we have*

$$\mathcal{E}_n^j \leq 2\Delta_{n-1}^j + 0.01\mathcal{E}_n^{j-1}.$$

Furthermore, for $j = 1$, $n = 1, \dots, N-1$, we have

$$\mathcal{E}_n^1 \leq 2\Delta_n^1 + 0.01 \left(\sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{t_n,\tau}^0(\omega) - \hat{\mathbf{y}}_{t_{n-1},\tau_{n-1,M}}^0(\omega) \right\|^2 \right] \right).$$

Proof. For each $\omega \in \Omega$ conditioned on the filtration \mathcal{F}_{t_n} , consider the auxiliary process defined as in the previous section,

$$d\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) + \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-1}(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n+\tau}(\omega),$$

and

$$d\hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) + \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-2}(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n+\tau}(\omega).$$

We have

$$\begin{aligned} & d \left(\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right) \\ &= \left[\frac{1}{2} \left(\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right) + \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-1}(\omega) \right) - \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-2}(\omega) \right) \right] d\tau. \end{aligned}$$

Then we can calculate the derivative $\frac{d}{d\tau} \left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2$ as

$$\begin{aligned} & \frac{d}{d\tau} \left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2 \\ &= 2 \left(\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right)^\top \left[\frac{1}{2} \left(\hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right) + \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-1}(\omega) \right) - \mathbf{s}_{t_n+g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau)}^{j-2}(\omega) \right) \right]. \end{aligned}$$

By integrating from 0 to τ , we have

$$\begin{aligned} & \left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2 - \left\| \hat{\mathbf{y}}_{t_n,0}^j(\omega) - \hat{\mathbf{y}}_{t_n,0}^{j-1}(\omega) \right\|^2 \\ &= \int_0^\tau \left\| \hat{\mathbf{y}}_{t_n,\tau'}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau'}^{j-1}(\omega) \right\|^2 d\tau' \\ & \quad + \int_0^\tau 2 \left(\hat{\mathbf{y}}_{t_n,\tau'}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau'}^{j-1}(\omega) \right)^\top \left[\mathbf{s}_{t_n+g_n(\tau')}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-1}(\omega) \right) - \mathbf{s}_{t_n+g_n(\tau')}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-2}(\omega) \right) \right] d\tau' \\ &\leq 2 \int_0^\tau \left\| \hat{\mathbf{y}}_{t_n,\tau'}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau'}^{j-1}(\omega) \right\|^2 d\tau' + \int_0^\tau \left\| \mathbf{s}_{t_n+g_n(\tau')}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-1}(\omega) \right) - \mathbf{s}_{t_n+g_n(\tau')}^\theta \left(\hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-2}(\omega) \right) \right\|^2 d\tau' \\ &\leq 2 \int_0^\tau \left\| \hat{\mathbf{y}}_{t_n,\tau'}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau'}^{j-1}(\omega) \right\|^2 d\tau' + L_s^2 \int_0^\tau \left\| \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-1}(\omega) - \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-2}(\omega) \right\|^2 d\tau'. \end{aligned}$$

By Theorem A.4, and $\hat{\mathbf{y}}_{t_n,0}^{j,p}(\omega) = \hat{\mathbf{y}}_{t_{n-1},\tau_{n-1},M}^j(\omega)$, we have

$$\left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2 \leq L_s^2 e^{2\tau} \int_0^\tau \left\| \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-1}(\omega) - \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-2}(\omega) \right\|^2 d\tau' + e^{2\tau} \Delta_{n-1}^j.$$

By taking expectation, for all $\tau \in [0, h_n]$

$$\begin{aligned} & \mathbb{E}_{\omega \sim \bar{p}|\mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n,\tau}^j(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) \right\|^2 - e^{2\tau} \Delta_{n-1}^j \\ &\leq L_s^2 e^{2\tau} \int_0^\tau \mathbb{E}_{\omega \sim \bar{p}|\mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-1}(\omega) - \hat{\mathbf{y}}_{t_n,g_n(\tau')}^{j-2}(\omega) \right\|^2 d\tau' \\ &\leq L_s^2 e^{2\tau} \tau \sup_{\tau' \in [0,\tau]} \mathbb{E}_{\omega \sim \bar{p}|\mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n,\tau'}^{j-1}(\omega) - \hat{\mathbf{y}}_{t_n,\tau'}^{j-2}(\omega) \right\|^2. \end{aligned}$$

Thus

$$\begin{aligned} & \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p}|\mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n,\tau}^{j-1}(\omega) - \hat{\mathbf{y}}_{t_n,\tau}^{j-2}(\omega) \right\|^2 \\ &\leq e^{2h_n} \Delta_{n-1}^j + L_s^2 e^{2h_n} h_n \mathcal{E}_n^{j-1}. \end{aligned}$$

For $j = 1$, we consider the following two processes,

$$d\hat{\mathbf{y}}_{t_n, \tau}^1(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n, \tau}^1(\omega) + \mathbf{s}_{t_n + g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_n, g_n(\tau)}^0(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n + \tau}(\omega),$$

and

$$d\hat{\mathbf{y}}_{t_n, \tau}^0(\omega) = \left[\frac{1}{2} \hat{\mathbf{y}}_{t_n, \tau}^0(\omega) + \mathbf{s}_{t_n + g_n(\tau)}^\theta \left(\hat{\mathbf{y}}_{t_{n-1}, \tau_{n-1}, M}^0(\omega) \right) \right] d\tau + d\mathbf{w}_{t_n + \tau}(\omega).$$

Similarly, we have

$$\begin{aligned} & \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n, \tau}^1(\omega) - \hat{\mathbf{y}}_{t_n, \tau}^0(\omega) \right\|^2 \\ & \leq e^{2h_n} \Delta_n^1 + L_s^2 e^{2h_n} h_n \left(\sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left\| \hat{\mathbf{y}}_{t_n, \tau}^0(\omega) - \hat{\mathbf{y}}_{t_{n-1}, \tau_{n-1}, M}^0(\omega) \right\|^2 \right). \end{aligned}$$

□

Lemma C.12 (One-step decomposition of Δ_n^j). Assume $L_s^2 e^{2h_n} h_n \leq 0.01$ and $e^{2h_n} \leq 2$. For any $j = 2, \dots, J$, $n = 1, \dots, N-1$, we have

$$\Delta_n^j \leq 3\Delta_{n-1}^j + 0.4\mathcal{E}_n^{j-1}.$$

Furthermore, for $j = 1$, $n = 1, \dots, N-1$, we have

$$\Delta_n^1 \leq 3\Delta_{n-1}^1 + 0.4 \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n, \tau}^0 - \hat{\mathbf{y}}_{n-1, \tau_{n-1}, M}^0 \right\|^2 \right].$$

For $n = 0$, we have $\Delta_0^j \leq 0.32\Delta_0^{j-1}$, and $\Delta_0^1 \leq \sup_{\tau \in [0, h_0]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_0}} \left[\left\| \hat{\mathbf{y}}_{t_0, \tau}^0(\omega) - \hat{\mathbf{y}}_{t_0, 0}^0(\omega) \right\|^2 \right]$.

Proof. By definition of $\hat{\mathbf{y}}_{t_n, \tau_{n, M}}^j(\omega)$ we have

$$\begin{aligned} & \left\| e^{-\frac{h_n}{2}} \hat{\mathbf{y}}_{t_n, \tau_{n, M}}^j - e^{-\frac{h_n}{2}} \hat{\mathbf{y}}_{t_n, \tau_{n, M}}^{j-1} \right\|^2 \\ & = \left\| \hat{\mathbf{y}}_{n, 0}^j - \hat{\mathbf{y}}_{n, 0}^{j-1} + \sum_{m'=0}^{m-1} e^{\frac{-\tau_{n, m'}+1}{2}} 2(e^{\epsilon_{n, m'}} - 1) \left[\mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-1}) - \mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-2}) \right] \right\|^2 \\ & \leq 2 \left\| \hat{\mathbf{y}}_{n, 0}^j - \hat{\mathbf{y}}_{n, 0}^{j-1} \right\|^2 + 2 \left\| \sum_{m'=0}^{m-1} e^{\frac{-\tau_{n, m'}+1}{2}} 2(e^{\epsilon_{n, m'}} - 1) \left[\mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-1}) - \mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-2}) \right] \right\|^2 \\ & \leq 2 \left\| \hat{\mathbf{y}}_{n, 0}^j - \hat{\mathbf{y}}_{n, 0}^{j-1} \right\|^2 + 32\epsilon_{n, m'}^2 M \sum_{m'=0}^{M-1} \left\| \left[\mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-1}) - \mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n, \tau_{n, m'}}^{j-2}) \right] \right\|^2 \\ & \leq 2 \left\| \hat{\mathbf{y}}_{n, 0}^j - \hat{\mathbf{y}}_{n, 0}^{j-1} \right\|^2 + 32h_n^2 \sup_{\tau \in [0, h_n]} L_s^2 \left\| \hat{\mathbf{y}}_{n, \tau}^{j-1} - \hat{\mathbf{y}}_{n, \tau}^{j-2} \right\|^2, \end{aligned}$$

where the second inequality is implied by that $e^x - 1 \leq 2x$ when $x < 1$. By taking expectation, and the assumption that $L_s^2 e^{2h_n} h_n \leq 0.1$ and $e^{2h_n} \leq 2$, we have

$$\begin{aligned} e^{-\frac{h_n}{2}} \Delta_n^j & = \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} e^{-\frac{h_n}{2}} \left[\left\| \hat{\mathbf{y}}_{t_n, \tau_{n, M}}^j - \hat{\mathbf{y}}_{t_n, \tau_{n, M}}^{j-1} \right\|^2 \right] \\ & \leq 2\mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n, 0}^j - \hat{\mathbf{y}}_{n, 0}^{j-1} \right\|^2 \right] + 32h_n^2 L_s^2 \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n, \tau}^{j-1} - \hat{\mathbf{y}}_{n, \tau}^{j-2} \right\|^2 \right] \\ & \leq 2\Delta_{n-1}^j + 0.32\mathcal{E}_n^{j-1}. \end{aligned}$$

Thus

$$\Delta_n^j \leq 3\Delta_{n-1}^j + 0.4\mathcal{E}_n^{j-1}.$$

In the remaining part, we will bound Δ_n^1 . By definition, we have

$$\begin{aligned} & \left\| e^{-\frac{h_n}{2}} \hat{\mathbf{y}}_{t_n, \tau_n, M}^1(\omega) - e^{-\frac{h_n}{2}} \hat{\mathbf{y}}_{t_n, \tau_n, M}^0(\omega) \right\|^2 \\ &= \left\| \hat{\mathbf{y}}_{n,0}^1 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 + \sum_{m'=0}^{m-1} e^{-\frac{\tau_{n, m'}+1}{2}} 2(e^{\epsilon_{n, m'}} - 1) \left[\mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n-1, \tau_{n, m'}}^0) - \mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n-1, \tau_n, M}^0) \right] \right\|^2 \\ &\leq 2 \left\| \hat{\mathbf{y}}_{n,0}^1 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2 + 2 \left\| \sum_{m'=0}^{M-1} e^{-\frac{\tau_{n, m'}+1}{2}} 2(e^{\epsilon_{n, m'}} - 1) \left[\mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n-1, \tau_{n, m'}}^0) - \mathbf{s}_{t_n + \tau_{n, m'}}^\theta(\hat{\mathbf{y}}_{n-1, \tau_n, M}^0) \right] \right\|^2 \\ &\leq 2 \left\| \hat{\mathbf{y}}_{n,0}^1 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2 + 32h_n^2 L_s^2 \sup_{\tau \in [0, h_n]} \left\| \hat{\mathbf{y}}_{n-1, \tau}^0 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2, \end{aligned}$$

where the second inequality is implied by that $e^x - 1 \leq 2x$ when $x < 1$. Thus with $L_s^2 e^{2h_n} h_n \leq 0.01$ and $e^{2h_n} \leq 2$, we have

$$\begin{aligned} e^{-\frac{h_n}{2}} \Delta_n^1 &= \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} e^{-\frac{h_n}{2}} \left[\left\| \hat{\mathbf{y}}_{t_n, \tau_n, M}^1 - \hat{\mathbf{y}}_{t_n, \tau_n, M}^0 \right\|^2 \right] \\ &\leq 2 \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n,0}^1 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2 \right] + 32h_n^2 L_s^2 \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n-1, \tau}^{1, P-1} - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2 \right] \\ &\leq 2\Delta_{n-1}^1 + 0.32 \sup_{\tau \in [0, h_n]} \mathbb{E}_{\omega \sim \bar{p} | \mathcal{F}_{t_n}} \left[\left\| \hat{\mathbf{y}}_{n, \tau}^0 - \hat{\mathbf{y}}_{n-1, \tau_n, M}^0 \right\|^2 \right]. \end{aligned}$$

□

Let $L_n^j = 2\Delta_{n-1}^j + 0.01\mathcal{E}_n^{j-1}$. We note that $L_n^j \geq \mathcal{E}_n^j$. Thus for $n \geq 1$ and $j \geq 2$,

$$\begin{aligned} L_n^j &= 2\Delta_{n-1}^j + 0.01\mathcal{E}_n^{j-1} \\ &\leq 2(80\Delta_{n-1}^j + 0.4\mathcal{E}_n^{j-1}) + 0.01L_n^j \\ &\leq 160\Delta_{n-1}^j + 0.01L_n^j. \end{aligned} \tag{33}$$

We recursively bound L_n^j as

$$L_n^j \leq \sum_{a=2}^n (0.01)^{j-2} 160^{n-a} \binom{n-a+j-2}{j-2} L_a^2 + \sum_{b=2}^j (0.01)^{j-b} 160^{n-1} \binom{n-1+j-b}{j-b} L_1^b.$$

Bound for $\sum_{a=2}^n (0.01)^{j-2} 160^{n-a} \binom{n-a+j-2}{j-2} L_a^2$. Firstly, we bound L_a^2 . To do so, by Lemma C.12, we bound Δ_n^1 as

$$\Delta_n^1 \leq 3\Delta_{n-1}^1 + 4\mathcal{E}_I \leq 3^n \Delta_0^1 + \sum_{i=0}^{n-1} 4 \cdot 3^i \mathcal{E}_I \leq 4 \sum_{i=0}^n 3^i \mathcal{E}_I \leq 3^{n+2} \mathcal{E}_I.$$

and by Lemma C.11, bound \mathcal{E}_n^1 as

$$\mathcal{E}_n^1 \leq 2\Delta_n^1 + 0.1\mathcal{E}_I \leq 3^{n+3} \mathcal{E}_I.$$

Furthermore, by Lemma C.12, we bound Δ_n^2 as

$$\Delta_n^2 \leq 3\Delta_{n-1}^2 + 0.4\mathcal{E}_n^1 \leq 3^n \Delta_0^2 + \sum_{i=0}^{n-1} 3^i \mathcal{E}_{n-i}^1 \leq 0.32 \cdot 3^n \mathcal{E}_I + 3^{n+3} n \mathcal{E}_I \leq 28 \cdot 3^n n \mathcal{E}_I.$$

Thus

$$L_a^2 = 2\Delta_{a-1}^2 + 0.01\mathcal{E}_a^1 \leq 28 \cdot 3^a a \mathcal{E}_I.$$

Furthermore, by $\binom{m}{n} \leq \left(\frac{em}{n}\right)^n$ for $m \geq n > 0$, we have

$$\begin{aligned} & \sum_{a=2}^n (0.01)^{j-2} 160^{n-a} \binom{n-a+j-2}{j-2} L_a^2 \\ & \leq (0.01)^{j-2} (28 \cdot 160^n n^2) e^{j-2} \left(\frac{n-a+j-2}{j-2}\right)^{j-2} \mathcal{E}_I \\ & \leq (e^2 \cdot 0.01)^{j-2} (28 \cdot 160^n n^2) \mathcal{E}_I. \end{aligned}$$

Bound for $\sum_{b=2}^j (0.01)^{j-b} 160^{n-1} \binom{n-1+j-b}{j-b} L_1^b$. By Lemma C.11, we have

$$\begin{aligned} \mathcal{E}_1^j & \leq 0.01\mathcal{E}_1^{j-1} + 2\Delta_0^j \\ & \leq (0.01)^j \mathcal{E}_I + \sum_{i=0}^{j-1} (0.01)^i 2\Delta_0^{j-i}. \end{aligned}$$

Combining the fact that $\Delta_0^j \leq 0.32^{j-1} \mathcal{E}_I$, we have

$$\mathcal{E}_1^j \leq 7 \cdot j \cdot 0.32^j \mathcal{E}_I.$$

Thus

$$\begin{aligned} L_1^b & = 2\Delta_0^b + 0.01\mathcal{E}_1^{b-1} \\ & \leq 2 \cdot 0.32^{b-1} \mathcal{E}_I + 0.01 \cdot 7 \cdot (b-1) \cdot 0.32^{b-1} \mathcal{E}_I \\ & \leq 7 \cdot b \cdot 0.32^{b-1} \mathcal{E}_I. \end{aligned}$$

Furthermore, by $\sum_{i=0}^m \binom{n+i}{n} x^i = \frac{1-(m+1)\binom{m+n+1}{n} B_x(m+1, n+1)}{(1-x)^{n+1}} \leq \frac{1}{(1-x)^{n+1}}$, we have

$$\begin{aligned} & \sum_{b=2}^j (0.01)^{j-b} 160^{n-1} \binom{n-1+j-b}{n-1} L_1^b \\ & \leq \sum_{b=2}^j (0.01)^{j-b} 160^{n-1} \binom{n-1+j-b}{n-1} 7 \cdot b \cdot 0.32^{b-1} \mathcal{E}_I \\ & \leq 22 \cdot 0.87^j 440^{n-1} j \mathcal{E}_I. \end{aligned}$$

Combining the above two results, we have

$$\mathcal{E}_n^J \leq (e^2 \cdot 0.01)^{j-2} (28 \cdot 160^n n^2) \mathcal{E}_I + 22 \cdot 0.87^j 440^{n-1} j \mathcal{E}_I.$$

If $J - 45N \gtrsim \log \frac{N\mathcal{E}_I}{\varepsilon^2}$, for any $n = 0, \dots, N$

$$\mathcal{E}_n^J \leq \frac{\varepsilon^2}{N}. \quad (34)$$

C.3.1. OVERALL ERROR BOUND

By the previous computation, we have

$$\begin{aligned} & \text{KL}(\tilde{p}_{t_{n+1}} \parallel \hat{q}_{t_{n+1}}) \\ & \leq \text{KL}(\tilde{p}_{t_n} \parallel \hat{q}_{t_n}) + \mathbb{E}_{\omega \sim q|_{\mathcal{F}_{t_n}}} \left[\frac{1}{2} \int_0^{h_n} \|\delta_{t_n}(\tau, \omega)\|^2 d\tau \right] \\ & \leq \text{KL}(\tilde{p}_{t_n} \parallel \hat{q}_{t_n}) + 3\mathbb{E}_{\omega \sim \tilde{p}|_{\mathcal{F}_{t_n}}} [A_{t_n}(\omega) + B_{t_n}(\omega)] + 3L_s^2 h_n \mathcal{E}_n^J. \end{aligned}$$

Combining Lemma A.5, Corollary C.10, and Eq. (34), we have

$$\begin{aligned}
 & \text{KL}(\tilde{p}_{t_{n+1}} \parallel \hat{q}_{t_{n+1}}) \\
 & \leq \text{KL}(\tilde{p}_0 \parallel \hat{q}_0) + 3 \sum_{n=0}^{N-1} \left(\mathbb{E}_{\omega \sim \tilde{p} | \mathcal{F}_{t_n}} [A_{t_n}(\omega) + B_{t_n}(\omega)] + L_{\mathbf{s}}^2 h_n \mathcal{E}_n^J \right) \\
 & \lesssim d e^{-T} + \epsilon d (T + \log \eta^{-1}) + \delta_2^2 + \varepsilon^2, \\
 & \lesssim \varepsilon^2,
 \end{aligned}$$

with parameters $J - 45N \geq \mathcal{O}(\log \frac{Nd}{\varepsilon^2})$, $h = \Theta(1)$, $N = \mathcal{O}(\log \frac{d}{\varepsilon^2})$, $T = \mathcal{O}(\log \frac{d}{\varepsilon^2})$, $\epsilon = \Theta(d^{-1} \varepsilon^2 \log^{-1} \frac{d}{\varepsilon^2})$, $M = \mathcal{O}(d \varepsilon^{-2} \log \frac{d}{\varepsilon^2})$, $\log \eta^{-1} \lesssim T$.