Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning

Ruqi Zhang[1] **Chunyuan Li**[2] **Jianyi Zhang**[3] **Changyou Chen**[4] **Andrew Gordon Wilson**[1]

Abstract

The posteriors over neural network weights are high dimensional and multimodal. Each mode typically characterizes a meaningfully different representation of the data. We develop Cyclical Stochastic Gradient MCMC (SG-MCMC) to automatically explore such distributions. In particular, we propose a cyclical stepsize schedule, where larger steps discover new modes, and smaller steps characterize each mode. We prove that our proposed learning rate schedule provides faster convergence to samples from a stationary distribution than SG-MCMC with standard decaying schedules. Moreover, we provide extensive experimental results to demonstrate the effectiveness of cyclical SG-MCMC in learning complex multimodal distributions, especially for fully Bayesian inference with modern deep neural networks.

1. Introduction

Deep neural networks are often trained with stochastic optimization methods such as stochastic gradient decent (SGD) and its variants. Bayesian methods provide a principled alternative, which account for model uncertainty in weight space (MacKay, 1992; Neal, 1996), and achieve an automatic balance between model complexity and data fitting. Indeed, Bayesian methods have been shown to improve the generalization performance of DNNs (Hernández-Lobato & Adams, 2015; Blundell et al., 2015; Li et al., 2016a), while providing a principled representation of uncertainty on predictions, which is crucial for decision making.

While a Bayesian approach ameliorates the over-fitting issue in these complicated models, exact Bayesian inference in DNNs is intractable. Approximate inference for Bayesian deep learning has typically focused on deterministic approaches, such as variational methods (Hernández-Lobato & Adams, 2015; Blundell et al., 2015). By contrast, MCMC methods are now essentially unused for inference with modern deep neural networks, despite previously providing

[1]Cornell University [2]Microsoft Research, Redmond [3]Fudan University [4]University at Buffalo, SUNY.

the gold standard of performance with smaller neural networks (Neal, 1996).

Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC) methods (Welling & Teh, 2011; Chen et al., 2014; Ding et al., 2014; Li et al., 2016a) provide a promising direction for a sampling based approach to inference in Bayesian deep learning. Indeed, it has been shown that stochastic methods, which use mini-batches of data, are crucial for finding weight parameters that provide good generalization in modern deep neural networks (Keskar et al., 2016).

However, SG-MCMC algorithms for inference with modern neural networks face several challenges: (*i*) In theory, SG-MCMC asymptotically converges to samples from target distributions via a decreasing stepsize scheme, but suffers from a bounded estimation error in limited time (Teh et al., 2016; Chen et al., 2015). (*ii*) In practice, empirical successes have been reported by training DNNs in relatively short time (Li et al., 2016b; Chen et al., 2014; Gan et al., 2016; Neelakantan et al., 2016; Saatchi & Wilson, 2017). For example, Saatchi & Wilson (2017) apply SG-MCMC to generative adversarial networks (GANs) to solve the mode collapse problem and capture diverse generation styles. However, the loss surface for DNNs is highly multimodal (Auer et al., 1996; Choromanska et al., 2015). In order for MCMC to be effective for posterior inference in modern neural networks, a crucial question remains: how do we make SG-MCMC efficiently explore a highly multimodal parameter space given a practical computational budget?

Several attempts have been made to improve the sampling efficiency of SG-MCMC. Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014) introduces the momentum variable to the Langevin dynamics. The preconditioned stochastic gradient Langevin dynamics (pS-GLD) (Li et al., 2016a) adaptively adjust the sampler's step size according to the local geometry of parameter space. Though simple and promising, these methods are still inefficient at exploring multimodal distributions in practice. It is our contention that this limitation arises from difficulties escaping local modes when using the small stepsizes that SG-MCMC methods typically require. Note that the stepsize in SG-MCMC controls the sampler's behavior in two ways: the magnitude to deterministically drift towards high density regions *wrt.* the current stochastic gradient, and the

level of injecting noise to randomly explore the parameter space. Therefore, a small stepsize reduces both abilities, resulting in a impractically large number of iterations required for the sampler to move across modes.

In this paper, we develop SG-MCMC methods specifically for exploring complex multimodal distributions, such as the posteriors over parameters for modern deep neural networks. In particular, we replace the traditional decreasing stepsize schedule in SG-MCMC with a cyclically changing one. To note the distinction from traditional SG-MCMC, we refer to this as the *Cyclical SG-MCMC* method (cSG-MCMC). The comparison is illustrated in Figure 1. The blue curve is the traditional decay, while the red curve shows the proposed cyclical schedule. Cyclical SG-MCMC operates in two stages: (*i*) *Exploration*: when the stepsize is large (dashed red curves), we consider this stage as an effective burn-in mechanism, encouraging the sampler to take large moves and leave the local mode using the stochastic gradient. (*ii*) *Sampling*: when the stepsize is small (solid red curves), the sampler explores one local mode. We collect samples for local distribution estimation during this stage. Further, we propose two practical techniques to improve the estimation efficiency: (1) a system temperature for exploration and exploitation; (2) a weighted combination scheme for samples collected in different cycles accommodate their relative importance.

This procedure can be viewed as SG-MCMC with warm restarts: the exploration stage provide the warm restarts for its following sampling stage. cSG-MCMC combines the advantages from (1) the traditional SG-MCMC to characterize the fine-scale local density distribution and (2) the cyclical schedule in optimization to efficiently explore multimodal of the parameter space. In limited time, cSG-MCMC is a practical tool to provide significantly better mixing than the traditional SG-MCMC for complex distributions. cSG-MCMC can also be considered as an *efficient* approximation to parallel MCMC; cSG-MCMC can achieve similar performance to parallel MCMC with only a fraction of cost (reciprocal to the number of chains) that parallel MCMC requires.

To support our proposal, we also show the non-asymptotic convergence for the cyclical schedule. We additionally show a faster convergence rate than for traditional decay schedules. We note that this is the first convergence analysis of a cyclical stepsize algorithm in general, including work that uses cyclical learning rates for different purposes in optimization (e.g., Loshchilov & Hutter, 2016; Huang et al., 2017; Garipov et al., 2018). Moreover, we provide extensive experimental results to demonstrate the advantages of cSG-MCMC in sampling from multimodal distributions, including a synthetic dataset, Bayesian logistic regression, Bayesian neural networks and uncertainty estimation.
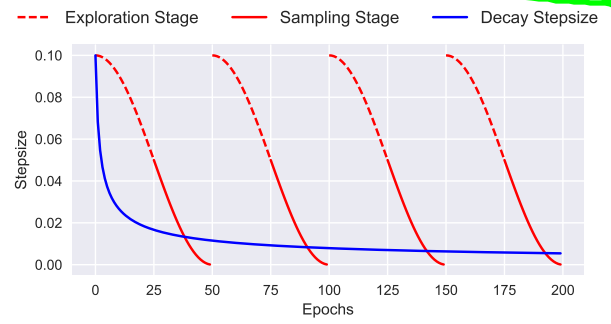

Figure 1. Illustration of the proposed cyclical stepsize schedule (red) and the traditional decreasing stepsize schedule (blue) for SG-MCMC algorithms.

In short, cSG-MCMC provides a simple and automatic approach to inference in modern Bayesian deep learning, with promising results, and theoretical support. This work is a step towards enabling MCMC approaches in Bayesian deep learning. We release the code on GitHub [1].

2. Preliminaries: SG-MCMC with a Decreasing Stepsize Schedule

SG-MCMC is a family of scalable sampling methods that enables inference with mini-batches of data. For a dataset $\mathcal{D} = \{d_i\}_{i=1}^N$ and a θ-parameterized model, we have the likelihood $p(\mathcal{D}|\theta)$ and prior $p(\theta)$. The posterior is

$$p(\theta|\mathcal{D}) \propto \exp(-U(\theta)),$$

where $U(\theta)$ is the potential energy given by

$$U(\theta) = -\log p(\mathcal{D}|\theta) - \log p(\theta).$$

When $\mathcal{D}$ is too large, it is expensive to evaluate $U(\theta)$ for all the data points at each iteration. Instead, SG-MCMC methods use a minibatch to approximate $U(\theta)$:

$$\tilde{U}(\theta) = -\frac{N'}{N} \sum_{i=1}^{N'} \log p(x_i|\theta) - \log p(\theta),$$

where $N' \ll N$ is the size of minibatch. We recommend (Ma et al., 2015) for a general review of SG-MCMC algorithms. We describe two SG-MCMC algorithms considered in this paper.

SGLD: Welling & Teh (2011) proposed Stochastic Gradient Langevin Dynamics (SGLD), which uses stochastic gradients and adds Gaussian noise. Posterior samples are updated at the k-th step as:

$$\theta_k = \theta_{k-1} - \alpha_k \nabla \tilde{U}(\theta_k) + \sqrt{2\alpha_k}\epsilon_k$$

---

[1] https://github.com/ruqizhang/csgmcmc

where α_k is the stepsize and ϵ_k has a standard Gaussian distribution.

SGHMC: To improve mixing over SGLD, Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC) (Chen et al., 2014) introduces an auxiliary momentum variable v. SG-HMC is built upon HMC, with an additional friction term to counteract the noise introduced by a mini-batch. The update rule for posterior samples is

$$\theta_k = \theta_{k-1} + v_{k-1}$$
$$v_k = v_{k-1} - \alpha_k \nabla \tilde{U}(\theta_k) - \eta v_{k-1} + \sqrt{2(\eta - \hat{\gamma})\alpha_k} \epsilon_k$$

where $1 - \eta$ is the momentum term and $\hat{\gamma}$ is the estimate of the noise.

To guarantee asymptotic consistency with the true distribution, SG-MCMC requires that the step sizes satisfy the following assumption:

Assumption 1. *The step sizes $\{\alpha_k\}$ are decreasing, i.e., $0 < \alpha_{k+1} < \alpha_k$, with 1) $\sum_{k=1}^{\infty} \alpha_k = \infty$; and 2) $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$.*

Without a decreasing step-size, the estimation error from numerical approximations is problematic. One typical decaying step-size schedule is

$$\alpha_k = a(b+k)^{-\gamma} \tag{1}$$

with $\gamma \in (0.5, 1]$ and (a, b) some positive constants (Welling & Teh, 2011).

3. Cyclical SG-MCMC

We now introduce our *cyclical SG-MCMC* (cSG-MCMC) algorithm. cSG-MCMC consists of two stages: *exploration* and *sampling*. In the following, we first introduce the cyclical step-size schedule, and then describe the exploration stage in Section 3.1 and the sampling stage in Section 3.2. We propose an approach to combining samples for testing in Section 3.3.

Assumption 1 guarantees the consistency of our estimation with the true distribution in asymptotic time. The approximation error in limited time is characterized as the risk of an estimator $R = B^2 + V$, where B is the bias and V is the variance. In the case of infinite computation time, the traditional SG-MCMC setting can reduce the bias and variance to zero. However, the time budget is often limited in practice, and there is always a trade-off between bias and variance. We therefore decrease the overall approximation error R by reducing the variance through obtaining more effective samples. The effective sample size can be increased if less correlated samples from different distribution modes are collected.

For deep neural networks, the parameter space is highly multimodal. SG-MCMC with the traditional decreasing stepsize schedule in practice becomes trapped in a local mode, though injecting noise may help the sampler to escape in the asymptotic regime (Zhang et al., 2017). Inspired to improve the exploration of the multimodal posteriors for deep neural networks, with a simple and automatic approach, we propose the cyclical cosine stepsize schedule for SG-MCMC. The stepsize at iteration k is defined as (Izmailov et al., 2018; Loshchilov & Hutter, 2016; Huang et al., 2017):

$$\alpha_k = \frac{\alpha_0}{2} \left[\cos\left(\frac{\pi \, \mathrm{mod}(k-1, \lceil K/M \rceil)}{\lceil K/M \rceil} \right) + 1 \right], \tag{2}$$

where α_0 is the initial stepsize, M is the number of cycles and K is the number of total iterations.

The stepsize α_k varies periodically with k. In each period, α_k starts at α_0, and gradually decreases to 0. Within one period, SG-MCMC starts with a large stepsize, resulting in aggressive exploration in the parameter space; as the stepsize is decreasing, SG-MCMC explores local regions. In the next period, the Markov chain restarts with a large stepsize, enforcing the sampler to escape from the current mode and explore a new area of the posterior. We note that we chose a significantly larger amplitude α_0 than would be typical in other contexts, since our purpose is to specifically find distinct modes.

Related work in optimization. In optimization, the cyclical cosine annealing stepsize schedule has been demonstrated to be able to find diverse solutions in multimodal objectives, though not specifically different modes, using stochastic gradient methods (Loshchilov & Hutter, 2016; Huang et al., 2017; Garipov et al., 2018). Alternatively, we adopt the technique to SG-MCMC as an effective scheme for sampling from multimodal distributions.

3.1. Exploration

The first stage of cyclical SG-MCMC, *exploration*, discovers parameters near local modes of an objective function. Unfortunately, it is undesirable to directly apply the cyclical schedule in optimization to SG-MCMC for collecting samples at every step. SG-MCMC often requires a small stepsize in order to control the error induced by the noise from using a minibatch approximation. If the stepsize is too large, the stationary distribution of SG-MCMC might be far away from the true posterior distribution. To correct this error, it is possible to do stochastic Metropolis-Hastings (MH) (Korattikara et al., 2014; Bardenet et al., 2014; Chen et al., 2016b). However, stochastic MH correction is still computationally too expensive. Further, it is easy to get rejected with an aggressive large stepsize, and every rejection is a waste of gradient computation.

To alleviate this problem, we propose to introduce a system temperature T to control the sampler's behavior:

$$p(\theta|\mathcal{D}) \propto \exp(-U(\theta)/T). \qquad (3)$$

Note that setting $T = 1$ corresponds to sampling from the true posterior. When $T \rightarrow 0$, the posterior distribution becomes a point mass. Sampling from $\lim_{T \rightarrow 0} \exp(-U(\theta)/T)$ is equivalent to minimizing $U(\theta)$; in this context, SG-MCMC methods become stochastic gradient optimization methods.

One may increase the temperature T from 0 to 1 when the step-size is decreasing. We simply consider $T = 0$ and perform optimization as the burn-in stage, when the completed proportion of a cycle $r(k) = \frac{\text{mod }(k-1, \lceil K/M \rceil)}{\lceil K/M \rceil}$ is smaller than a given threshold: $r(k) < \beta$. Note that $\beta \in (0, 1)$ balances the proportion of the exploration and sampling stages in cSG-MCMC.

3.2. Sampling

The *sampling* stage corresponds to $T = 1$ of the exploration stage. When $r(k) > \beta$, or the step-sizes are sufficiently small, we initiate SG-MCMC updates and collect samples until this cycle ends.

SG-MCMC with Warm Restarts   One may consider the exploration stage as automatically providing warm restarts for the sampling stage. Exploration alleviates the inefficient mixing and inability to traverse the multimodal distributions of the traditional SG-MCMC methods. SG-MCMC with warm restarts explores different parts of the posterior distribution and capture multiple modes in a single training procedure.

In summary, the proposed cyclical SG-MCMC methods repeat the *two* stages, with three key advantages: (*i*) It restarts with a large stepsize at the beginning of a cycle which provides enough perturbation and encourages the model to escape from the current mode. (*ii*) The stepsize decreases more quickly inside one cycle than a traditional schedule, making the sampler better characterize the density of the local regions. (*iii*) This cyclical stepsize shares the advantage of the "super-convergence" property discussed in (Smith & Topin, 2017): cSG-MCMC can accelerate convergence for DNNs by up to an order of magnitude.

We emphasize that cyclical SG-MCMC still samples from the true posterior asymptotically, because we only collect samples when $T = 1$ in the sampling stage. Moreover, the cyclical schedule accelerates convergence, as shown in the non-asymptotic bound of Section 4.

Connection to the Santa algorithm.   It is interesting to note that our approach inverts steps of the Santa algorithm

(Chen et al., 2016a) for optimization. Santa is a simulated-annealing-based optimization algorithm with an exploration stage when $T = 1$, then gradually anneals $T \rightarrow 0$ in a refinement stage for global optimization. In contrast, our goal is to draw samples for multimodal distributions, thus we explore with $T = 0$ and sample with $T = 1$. Another fundamental difference is that Santa adopts the traditional stepsize decay, while we use the cyclical schedule.

We visually compare the difference between cyclical and traditional step size schedules in Figure 1. Eq. (1) specifies the traditional schedule. We present our cyclical SG-MCMC in Algorithm 1.

Algorithm 1 Cyclical SG-MCMC.

Input: The initial stepsize α_0, number of cycles M, number of training iterations K and the proportion of exploration stage β.

    for k = 1:K **do**

        $\alpha \leftarrow \alpha_k$ according to Eq (2).

        if $\frac{\text{mod }(k-1, \lceil K/M \rceil)}{\lceil K/M \rceil} < \beta$ **then**

            % Exploration stage

            $\theta \leftarrow \theta - \alpha \nabla \tilde{U}_k(\theta)$

        else

            % Sampling stage

            Collect samples using SG-MCMC methods

Output: Samples $\{\theta_k\}$

3.3. Combining Samples

In cyclical SG-MCMC, we obtain samples from multiple modes of a posterior distribution by running the cyclical step size schedule for many periods. We now show how to effectively utilize the collected samples.

We consider each cycle exploring different part of the target distribution $p(\theta|\mathcal{D})$ on a metric space Θ. As we have M cycles in total, the mth cycle characterizes a local region $\Theta_m \subset \Theta$, defining the "sub-posterior" distribution:

$$p_m(\theta|\mathcal{D}) = \frac{p(\theta|\mathcal{D})\mathbf{1}_{\Theta_m}}{w_m}, \text{ with } w_m = \int_{\Theta_m} p(\theta|\mathcal{D})d\theta,$$

where w_m is a normalizing constant.

For a testing function $f(\theta)$, we are often interested in its true posterior expectation $\bar{f} = \int f(\theta)p(\theta|\mathcal{D})d\theta$. The sample-based estimation is:

$$\hat{f} = \sum_{m=1}^{M} w_m \hat{f}_m \text{ with } \hat{f}_m = \frac{1}{K_m} \sum_{j=1}^{K_m} f(\theta_j^{(m)}), \qquad (4)$$

where K_m is the number of samples from the mth cycle, and $\theta^{(m)} \in \Theta_m$.

The weight for each cycle w_i is estimated using the *harmonic mean* method (Green, 1995; Raftery et al., 2006):

$\hat{w}_m \approx [\frac{1}{K_m} \sum_{j=1}^{K_m} \frac{1}{p(\mathcal{D}|\theta_j^{(m)})}]^{-1}$. This approach provides a simple and consistent estimator, where the only additional cost is to traverse the training dataset to evaluate the likelihood $p(\mathcal{D}|\theta_j^{(m)})$ for each sample $\theta_j^{(m)}$. We evaluate the likelihood once off-line and store the result for testing.

Connection to Parallel MCMC. Running parallel Markov chains is a natural and effective way to draw samples from multimodal distributions (VanDerwerken & Schmidler, 2013; Ahn et al., 2014). However, the training cost increases linearly with the number of chains. Cyclical SG-MCMC can be seen as an efficient way to approximate parallel MCMC. Each cycle effectively estimates a different region of posterior. Note cyclical SG-MCMC runs along a single training pass. Therefore, its computational cost is the same as single chain SG-MCMC while significantly less than parallel MCMC.

4. Theoretical Analysis

In this section, we show superior convergence rates for the proposed cSG-MCMC framework with a cyclical stepsize sequence $\{\alpha_k\}$ defined in (2) over standard SG-MCMC methods with traditional decreasing stepsizes $h_k = \frac{h_0}{k}$.[^2] Following existing work, we consider the case of SGLD and characterize convergence rates in terms of the 2-Wasserstein distance defined as:

$$W_2^2(\mu, \nu) := \inf_\gamma \left\{ \int_{\Omega \times \Omega} \|\theta - \theta'\|_2^2 \mathrm{d}\gamma(\theta, \theta') : \gamma \in \Gamma(\mu, \nu) \right\}$$

where $\Gamma(\mu, \nu)$ is the set of joint distributions over (θ, θ') such that the two marginals equal μ and ν, respectively. For simplicity, we do not consider the exploration stage in the analysis as that corresponds to stochastic optimization. For a fair comparison, we consider both algorithms running for the same number of iterations.

Our algorithm is based on the SDE characterizing the Langevin dynamics: $\mathrm{d}\theta_t = -\nabla U(\theta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathcal{W}_t$, where $\mathcal{W}_t \in \mathbb{R}^d$ is a d-dimensional Brownian motion. Denote the distribution of θ_t as ν_t. According to Chiang & Hwang (1987), the stationary distribution ν_∞ matches our target distribution. Let μ_K be the distribution of the sample from our proposed cSGLD algorithm at the K-th iteration, and $\tilde{\mu}_K$ the one from SGLD with polynomially decaying stepsizes. Our goal is to compare the convergence accuracies in terms of $W_2(\mu_K, \nu_\infty)$ and $W_2(\tilde{\mu}_T, \nu_\infty)$. We adopt standard assumptions as in most existing work, which are detailed in Assumption 2 in the Appendix. Theorem 1 summarizes our main theoretical result.

[^2]: Similar results hold for the more general decaying stepsizes $h_k = a(b + k)^\gamma$ with $\gamma \in (0.5, 1]$ defined in (Welling & Teh, 2011) We adopt $h_k = \frac{h_0}{k}$ here for simplicity.

Theorem 1. *Under Assumption 2 in the Appendix, there exist constants* $(C_0, C_1, C_2, C_3, C_4)$ *independent of the stepsizes such that the convergence rate of our proposed cSGLD with cyclical stepsize sequence* (2) *is bounded for all K satisfying $(K \bmod M = 0)$, as*

$$W_2(\mu_K, \nu_\infty) \leq W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k}) + W_2(\nu_{\sum_{k=1}^K \alpha_k}, \nu_\infty) \tag{5}$$

where $W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k}) \leq \left(6 + \frac{C_2 K \alpha_0}{2}\right)^{\frac{1}{2}} \times$

$$[(C_1 \frac{3\alpha_0^2 K}{8} + \sigma C_0 \frac{K\alpha_0}{2})^{\frac{1}{2}} + (C_1 \frac{3\alpha_0^2 K}{16} + \sigma C_0 \frac{K\alpha_0}{4})^{\frac{1}{4}}]$$

and $W_2(\nu_{\sum_{k=1}^K \alpha_k}, \nu_\infty) \leq C_3 \exp(-\frac{K\alpha_0}{2C_4})$. *For the standard polynomially-decaying-stepsize SGLD, the convergence rate is bounded as*

$$W_2(\tilde{\mu}_K, \nu_\infty) \leq W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^K h_k}) + W_2(\nu_{\sum_{k=1}^K h_k}, \nu_\infty), \tag{6}$$

where $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^K h_k}) \leq (6 + h_0 \sum_{k=1}^K \frac{1}{k})^{\frac{1}{2}} \times$

$$[(C_1 h_0^2 \frac{\pi^2}{6} + \sigma C_0 h_0 \sum_{k=1}^K \frac{1}{k})^{\frac{1}{2}} + (C_1 h_0^2 \frac{\pi^2}{16} + \sigma C_0 \frac{h_0}{2} \sum_{k=1}^K \frac{1}{k})^{\frac{1}{4}}]$$

and $W_2(\nu_{\sum_{k=1}^K h_k}, \nu_\infty) \leq C_3 \exp(-\frac{\sum_{k=1}^K h_k}{C_4})$.

To compare convergence rates of our algorithm cSGLD with standard SGLD, consider the following two cases:

i) If the initial stepsizes satisfy $\alpha_0 \geq h_0$, our algorithm cSGLD runs much faster than the standard SGLD in terms of the amount of "diffusion time" *i.e.*, the t indexing θ_t in the continuous-time SDE mentioned above. This result follows from $\sum_{k=1}^K \alpha_k = \frac{K\alpha_0}{2}$ and $\sum_{k=1}^K h_k = \sum_{k=1}^K \frac{h_0}{k} = \mathcal{O}(h_0 \log K) \ll \frac{K\alpha_0}{2}$. In standard SGLD, in order to reduce the error described by $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^K h_k})$ (increases w.r.t. K), h_0 needs to be set small enough in practice, *e.g.*, $h_0 = 0.01$ in Welling & Teh (2011). Following the general analysis of SGLD in Raginsky et al. (2017) and Xu et al. (2017), the dominant term in the decomposition (6) is $W_2(\nu_{\sum_{k=1}^K h_k}, \nu_\infty)$ since it decreases exponentially fast with the increase of t. This result applies to the decomposition (5) for our algorithm. Since $\sum_{k=1}^K \alpha_k$ increases much faster in our algorithm than the term $\sum_{k=1}^K h_k$ in standard SGLD, our algorithm thus endows less error for K iterations, *i.e.*, $W_2(\nu_{\sum_{k=1}^K \alpha_k}, \nu_\infty) \ll W_2(\nu_{\sum_{k=1}^K h_k}, \nu_\infty)$. Hence, our algorithm outperforms standard SGLD, as will be verified in our experiments.

ii) One may consider increasing h_0 to make standard SGLD run as "fast" as our proposed algorithm, *i.e.*, $\sum_{k=1}^K h_k \approx \sum_{k=1}^K \alpha_k$. In this case, it is worth noting that h_0 scales as $\mathcal{O}(\alpha_0 K / \log K)$. Thus
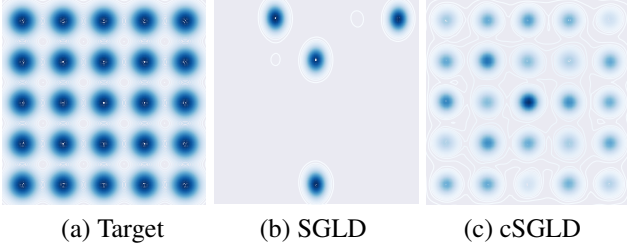
(a) Target   (b) SGLD   (c) cSGLD

Figure 2. Sampling from a mixture of 25 Gaussians shown in (a) for the parallel setting. With a budget of 50k×4=200k samples, traditional SGLD in (b) has only discovered 4 of the 25 modes, while our cSGLD in (c) has fully explored the distribution.

the h_0^2 term in $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^K h_k})$ would scale as $\mathcal{O}(\alpha_0^2 K^2/\log^2 K)$, which makes $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^K \alpha_k})$ in (6) much larger than our $W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k})$ defined in (5) since $\mathcal{O}(\alpha_0^2 K^2/\log^2 K) \gg \mathcal{O}(\alpha_0^2 K)$. Again, our algorithm cSGLD achieves a faster convergence rate than standard SGLD.

5. Experiments

We demonstrate cSG-MCMC on several tasks, including a synthetic multimodal distribution (Section 5.1), Bayesian logistic regression (Section 5.2) and image classification and uncertainty estimation with Bayesian neural networks (Sections 5.3 and 5.4). We choose SLGD and SGHMC as the representative baseline algorithms. Their cyclical counterpart are called cSGLD and cSGHMC, respectively.

5.1. Synthetic multimodal data

We first demonstrate the ability of cSG-MCMC for sampling from a multi-modal distribution on a 2D mixture of 25 Gaussians. Specifically, we compare cSGLD with SGLD in two setting: (1) parallel running with 4 chains and (2) running with a single chain, respectively. Each chain runs for 50k iterations. The stepsize schedule of SGLD is $\alpha_k \propto 0.05 k^{-0.55}$. In cSGLD, we set $M = 30$ and the initial stepsize $\alpha_0 = 0.09$. The proportion of exploration stage $\beta = \frac{1}{4}$. Fig 2 shows the estimated density using sampling results for SGLD and cSGLD in the parallel setting. We observed that SGLD gets trapped in the local modes, depending on the initial position. In any practical time period, SGLD could only characterize partial distribution. In contrast, cSGLD is able to find and characterize all modes, regardless of the initial position. cSGLD leverages large step sizes to discover a new mode, and small step sizes to explore local modes. This result suggests cSGLD can be a significantly favourable choice in the non-asymptotic setting, for example only 50k iterations in this case. The single chain results and the quantitative results on mode converge are reported in Section A.1 of Appendix.

5.2. Bayesian Logistic Regression

We consider Bayesian logistic regression (BLR) on three real-world datasets from the UCI repository: *Australian* (15 covariates, 690 data points), *German* (25 covariates, 1000 data points) and *Heart* (14 covariates, 270 data points). For all experiments, we collect 5000 samples with 1000 burn-in iterations. Following the settings in Li et al. (2016a), we report median *effective sample size* (ESS) in Table 1.

Note that BLR is *unimodal* in parameter space. We use this experiment as an adversarial situation for cSG-MCMC, which we primarily designed to explore multiple modes. Even in the *unimodal* setting, cSG-MCMC more effectively explores the parameter space than popular alternatives, in accordance with our Theorem 1. We can also use these experiments to understand how samplers respond to varying parameter dimensionality and training set sizes.

Overall, cSG-MCMC dramatically outperforms SG-MCMC, which demonstrates the fast mixing rate due to the warm restarts. On the small dataset *Heart*, SGHMC and cSGHMC achieve the same results, because the posterior of BLR on this dataset is simple. However, in higher dimensional spaces (*e.g., Australian* and *German*), cSG-MCMC shows significantly higher ESS; this result means that each cycle in cSG-MCMC can characterize a different region of the posterior, and combining multiple cycles yields more accurate overall approximation. These results support our Theorem 1, showing that cSG-MCMC is converges faster than conventional SG-MCMC, regardless of the number of modes of the distribution.

	Australian	**German**	**Heart**
SGLD	1676	492	2199
cSGLD	2138	978	2541
SGHMC	1317	2007	**5000**
cSGHMC	**4707**	**2436**	**5000**

Table 1. Effective sample size for samples for the unimodal posteriors in Bayesian linear regression, obtained using cyclical and traditional SG-MCMC algorithms, respectively.

5.3. Bayesian Neural Networks

We demonstrate the effectiveness of cSG-MCMC on Bayesian neural networks for classification on CIFAR-10 and CIFAR-100. We compare with (*i*) traditional SG-MCMC; (*ii*) traditional stochastic optimization methods, including stochastic gradient descent (SGD) and stochastic gradient descent with momentum (SGDMOM); and (*iii*) *Snapshot*: a stochastic optimization ensemble method method with a the cyclical stepsize schedule (Huang et al., 2017). We use a ResNet-18 (He et al., 2016) and run all algorithms for 200 epochs. We report the test errors averaged over 3 runs, and the standard error (±) from the mean

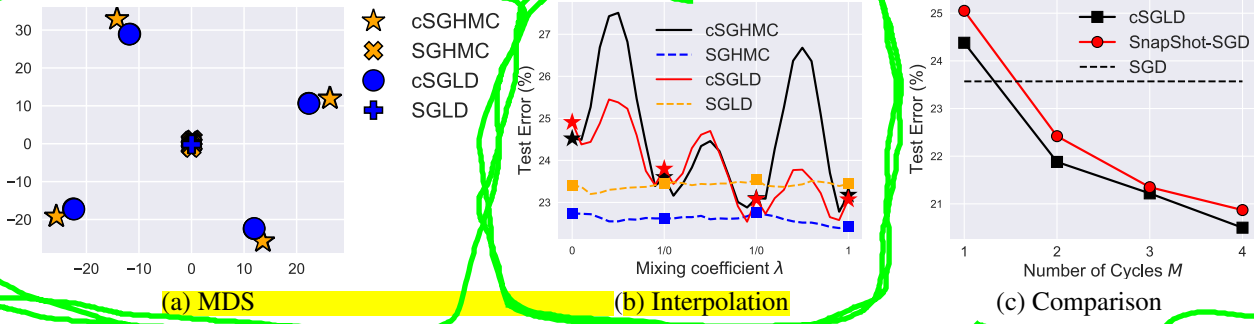(a) MDS   (b) Interpolation   (c) Comparison

Figure 3. Results of cSG-MCMC with DNNs on CIFAR-100 dataset. (a) MDS visualization in weight space: cSG-MCMC show larger distance than traditional schedules. (b) Testing errors (%) on the path of two samples: cSG-MCMC shows more varied performance. (c) Testing errors (%) as a function of the number of cycles M: cSGLD yields consistently lower errors.

	CIFAR-10	CIFAR-100
SGD	5.29±0.15	23.61±0.09
SGDMOM	5.17±0.09	22.98±0.27
Snapshot-SGD	4.46±0.04	20.83±0.01
Snapshot-SGDMOM	4.39±0.01	20.81±0.10
SGLD	5.20±0.06	23.23±0.01
cSGLD	4.29±0.06	20.55±0.06
SGHMC	4.93±0.1	22.60±0.17
cSGHMC	**4.27**±0.03	**20.50**±0.11

Table 2. Comparison of test error (%) over three runs (with ± standard error) between sSG-MCMC with non-parallel algorithms on CIFAR-10 and CIFAR-100. cSGLD and cSGHMC yields lower errors than their optimization counterparts, respectively.

predictor.

We set $M = 4$ and $\alpha_0 = 0.5$ for cSGLD, cSGHMC and Snapshot. The proportion hyper-parameter $\beta = 0.8$ and 0.94 for CIFAR-10 and CIFAR-100, respectively. We collect 3 samples per cycle. In practice, we found that the collected samples share similarly high likelihood for DNNs, thus one may simply set the normalizing term w_i in Eq. (4) to be the same for faster testing.

For the traditional SG-MCMC methods, we found that noise injection early in training hurts convergence. To make these baselines as competitive as possible, we thus avoid noise injection for the first 150 epochs of training (corresponding to the zero temperature limit of SGLD and SGHMC), and resume SGMCMC as usual (with noise) for the last 50 epochs. This scheme is similar to the exploration and sampling stages within one cycle of cSG-MCMC. We collect 20 samples for the MCMC methods and average their predictions in testing.

Testing Performance for Image Classification We report the testing errors in Table 2 to compare with the non-parallel algorithms. Snapshot and traditional SG-MCMC

reduce the testing errors on both datasets. Performance variance for these methods is also relatively small, due to the multiple networks in the Bayesian model average. Further, cSG-MCMC significantly outperforms Snapshot ensembles and the traditional SG-MCMC, which demonstrates the importance of (1) capturing diverse modes compared to traditional SG-MCMC, and (2) capturing fine-scale characteristics of the distribution compared with Snapshot ensembles.

Diversity in Weight Space. To further demonstrate our hypothesis that with a limited budget cSG-MCMC can find diverse modes, while traditional SG-MCMC cannot, we visualize the 12 samples we collect from cSG-MCMC and SG-MCMC on CIFAR-100 respectively using Multidimensional Scaling (MDS) in Figure 3 (a). MDS uses a Euclidean distance metric between the weight of samples. We see that the samples of cSG-MCMC form 4 clusters, which means they are from 4 different modes in weight space. However, all samples from SG-MCMC only form one cluster, which indicates traditional SGHMC gets trapped in one mode and only samples from that mode.

Diversity in Prediction. To further demonstrate that the samples from different cycles of cSG-MCMC provide diverse predictions we choose one sample from each cycle and linearly interpolate between two of them (Goodfellow et al., 2014; Huang et al., 2017). Specifically, let $J(\theta)$ be the test error of a sample with parameter θ. We compute the test error of the convex combination of two samples $J(\lambda\theta_1 + (1-\lambda)\theta_2)$, where $\lambda \in [0, 1]$.

We linearly interpolate between two samples from neighboring chains of cSG-MCMC since they are the most likely to be similar. We randomly select 4 samples from SG-MCMC. If the samples are from the same mode, the test error of the linear interpolation of parameters will be relatively smooth, while if the samples are from different modes, the test error of the parameter interpolation will have a spike when λ is

Method	Cyclical+Parallel		Decreasing+Parallel		Decreasing+Parallel		Cyclical+Single	
Cost	200/800		200/800		100/400		200/200	
Sampler	SGLD	SGHMC	SGLD	SGHMC	SGLD	SGHMC	SGLD	SGHMC
CIFAR-10	4.09	3.95	4.15	4.09	5.11	4.52	4.29	4.27
CIFAR-100	19.37	19.19	20.29	19.72	21.16	20.82	20.55	20.50

Table 3. Comparison of test error (%) between cSG-MCMC with parallel algorithm (M=4 chains) on CIFAR-10 and CIFAR-100. The method is reported in the format of "step-size schedule (cyclical or decreasing) + single/parallel chain". The cost is reported in the format of "#epoch per chain / #epoch used in all chains". Note that a parallel algorithm with a single chain reduces to a non-parallel algorithm. Integration of the cyclical schedule with parallel algorithms provides lower testing errors.

between 0 and 1.

We show the results of interpolation for cSG-MCMC and SG-MCMC on CIFAR-100 in Figure 3 (b). We see a spike in the test error in each linear interpolation of parameters between two samples from neighboring chains in cSG-MCMC while the linear interpolation for samples of SG-MCMC is smooth. This result suggests that samples of cSG-MCMC from different chains are from different modes while samples of SG-MCMC are from the same mode.

Although the test error of a single sample of cSG-MCMC is worse than that of SG-MCMC shown in Figure 3 (c), the ensemble of these samples significantly improves the test error, indicating that samples from different modes provide different predictions and make mistakes on different data points. Thus these diverse samples can complement each other, resulting in a lower test error, and demonstrating the advantage of exploring diverse modes using cSG-MCMC.

Comparison to Parallel MCMC. cSG-MCMC can be viewed as an economical alternative to parallel MCMC. We verify how closely cSG-MCMC can approximate the performance of parallel MCMC, but with more convenience and less computational expense. We also note that we can improve parallel MCMC with the proposed cyclical stepsize schedule.

We report the testing errors in Table 3 to compare multiple-chain results. (1) Four chains used, each runs 200 epochs (800 epochs in total), the results are shown in the first 4 columns (Cyclical+Parallel vs Decreasing+Parallel). We see that cSG-MCMC variants provide lower errors than plain SG-MCMC. (2) We reduce the number of epochs (#epoch) of parallel MCMC to 100 epoch each for decreasing stepsize schedule. The total cost is 400 epochs. We compare its performance with cyclical single chain (200 epochs in total) in the last 4 columns (Decreasing+Parallel vs Cyclical+Single). We see that the cyclical schedule running on a single chain performs best even with half the computational cost! All the results indicate the importance of warm re-starts using the proposed cyclical schedule. For a given total cost budget for both methods, the proposed cSGMCMC is preferable to parallel sampling.

Comparison to Snapshot Optimization. We carefully compared with Snapshot, as our cSG-MCMC can be viewed as the sampling counterpart of the Snapshot optimization method. We plot the test error wrt various number of cycles M in Fig. 3.

As M increases, cSG-MCMC and Snapshot both improve. However, given a fixed M, cSG-MCMC yields substantially lower test errors than Snapshot. This result is due to the ability of sSG-MCMC to better characterize the local distribution of modes: Snapshot provides a singe minima per cycle, while cSG-MCMC fully exploits the mode with more samples, which could provide weight uncertainty estimate and avoid over-fitting.

5.4. Uncertainty Evaluation

To demonstrate how predictive uncertainty benefits from exploring multiple modes in the posterior of neural network weights, we consider the task of uncertainty estimation for out-of-distribution samples (Lakshminarayanan et al., 2017). We train a three-layer MLP model on the standard MNIST train dataset until convergence using different algorithms, and estimate the entropy of the predictive distribution on the notMNIST dataset (Bulatov, 2011). Since the samples from the notMNIST dataset belong to the unseen classes, ideally the predictive distribution of the trained model should be uniform over the notMNIST digits, which gives the maximum entropy.

In Figure 4, we plot the empirical CDF for the entropy of the predictive distributions on notMNIST. We see that the uncertainty estimates from cSGHMC and cSGLD are better than the other methods, since the probability of a low entropy prediction is overall lower. cSG-MCMC algorithms explore more modes in the weight space, each mode characterize a meaningfully different representation of MNIST data. When testing on the out-of-distribution dataset (notMNIST), each mode can provide different predictions over the label space, leading to more reasonable uncertainty estimates. The Snapshot method achieves less entropy than cSG-MCMC, since it represents each mode with a single point.
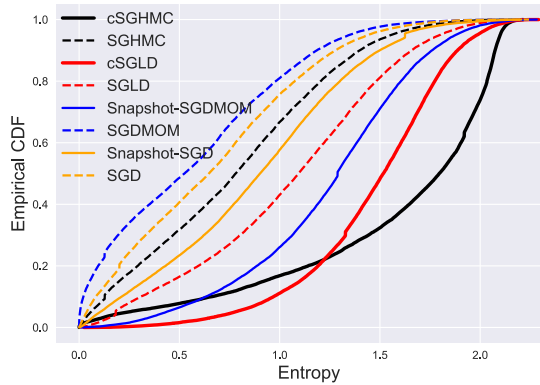
Figure 4. Empirical CDF for the entropy of the predictive distribution on notMNIST dataset. cSGLD and cSGHMC show lower probability for the low entropy estimate than other algorithms.

The traditional SG-MCMC methods also provide better uncertainty estimation compared to their optimization counterparts, because they characterize a local region of the parameter space, rather than a single point. cSG-MCMC can be regarded as a combination of these two worlds: a wide coverage of many modes in Snapshot, and fine-scale characterization of local regions in SG-MCMC.

6. Discussion

We have proposed cyclical SG-MCMC methods to automatically explore complex multimodal distributions. Our approach is particularly compelling for Bayesian deep learning, which involves rich multimodal parameter posteriors corresponding to meaningfully different representations. We have also shown that our cyclical methods explore unimodal distributions more efficiently. These results are in accordance with theory we developed to show that cyclical SG-MCMC will converge faster to samples from a stationary distribution in general settings. Moreover, we show cyclical SG-MCMC methods provide more accurate uncertainty estimation, by capturing more diversity in the hypothesis space corresponding to settings of model parameters.

While MCMC was once the gold standard for inference with neural networks, it is now rarely used in modern deep learning. We hope that this paper will help renew interest in MCMC for this purpose. Indeed, MCMC is uniquely positioned to explore the rich multimodal posterior distributions of modern neural networks, which can lead to improved accuracy, reliability, and uncertainty representation.

Acknowledgements

References

Ahn, S., Shahbaba, B., and Welling, M. Distributed stochastic gradient mcmc. In *ICML*, 2014.

Auer, P., Herbster, M., and Warmuth, M. K. Exponentially many local minima for single neurons. In *NIPS*, 1996.

Bardenet, R., Doucet, A., and Holmes, C. Towards scaling up markov chain monte carlo: an adaptive subsampling approach. In *ICML*, 2014.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *ICML*, 2015.

Bolley, F. and Villani, C. Weighted csiszár-kullback-pinsker inequalities and applications to transportation inequalities. In *Annales de la Faculte des sciences de Toulouse*. Université Paul Sabatier, 2005.

Bulatov, Y. Not MNIST Dataset. 2011. http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html.

Chen, C., Ding, N., and Carin, L. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *NIPS*, 2015.

Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In *Artificial Intelligence and Statistics*, 2016a.

Chen, H., Seita, D., Pan, X., and Canny, J. An efficient minibatch acceptance test for metropolis-hastings. *arXiv preprint arXiv:1610.06848*, 2016b.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *ICML*, 2014.

Chiang, T.-S. and Hwang, C.-R. Diffusion for global optimization in rn. *SIAM J. Control Optim.*, pp. 737–753, 1987. ISSN 0363-0129. doi: 10.1137/0325042. URL http://dx.doi.org/10.1137/0325042.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, 2015.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. Bayesian sampling using stochastic gradient thermostats. In *NIPS*, 2014.

Gan, Z., Li, C., Chen, C., Pu, Y., Su, Q., and Carin, L. Scalable bayesian learning of recurrent neural networks for language modeling. *ACL*, 2016.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

Green, P. J. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *ICML*, 2015.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get m for free. *ICLR*, 2017.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Korattikara, A., Chen, Y., and Welling, M. Austerity in mcmc land: Cutting the metropolis-hastings budget. In *ICML*, 2014.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.

Li, C., Chen, C., Carlson, D. E., and Carin, L. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, 2016a.

Li, C., Stevens, A., Chen, C., Pu, Y., Gan, Z., and Carin, L. Learning weight uncertainty with stochastic gradient mcmc for shape classification. In *CVPR*, 2016b.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. 2016.

Ma, Y.-A., Chen, T., and Fox, E. A complete recipe for stochastic gradient mcmc. In *NIPS*, 2015.

MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 1992.

Neal, R. M. *Bayesian learning for neural networks*. New York: Springer-Verlag, 1996.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. Adding gradient noise improves learning for very deep networks. *ICLR workshop*, 2016.

Raftery, A. E., Newton, M. A., Satagopan, J. M., and Krivitsky, P. N. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. 2006.

Raginsky, M., Rakhlin, A., and Telgarsky, M. Nonconvex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*, 2017.

Saatchi, Y. and Wilson, A. G. Bayesian GAN. *NIPS*, 2017.

Smith, L. N. and Topin, N. Super-convergence: Very fast training of residual networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2017.

Teh, Y. W., Thiery, A. H., and Vollmer, S. J. Consistency and fluctuations for stochastic gradient langevin dynamics. *The Journal of Machine Learning Research*, 2016.

VanDerwerken, D. N. and Schmidler, S. C. Parallel markov chain monte carlo. *arXiv preprint arXiv:1312.7479*, 2013.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.

Xu, P., Chen, J., Zou, D., and Gu, Q. Global convergence of langevin dynamics based algorithms for nonconvex optimization. *arXiv preprint arXiv:1707.06618*, 2017.

Zhang, J., Zhang, R., and Chen, C. Stochastic particle-optimization sampling and the non-asymptotic convergence theory.

Zhang, Y., Liang, P., and Charikar, M. A hitting time analysis of stochastic gradient langevin dynamics. In *COLT*, 2017.

Appendix:
Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning

A. Experimental Results

A.1. Synthetic Multimodal Distribution

The density of the distribution is

$$F(x) = \sum_{i=1}^{25} \lambda \mathcal{N}(x|\mu_i, \Sigma),$$

where $\lambda = \frac{1}{25}$, $\mu = \{-4, -2, 0, 2, 4\}^\top \times \{-4, -2, 0, 2, 4\}$, $\Sigma = \begin{bmatrix} 0.03 & 0 \\ 0 & 0.03 \end{bmatrix}$.

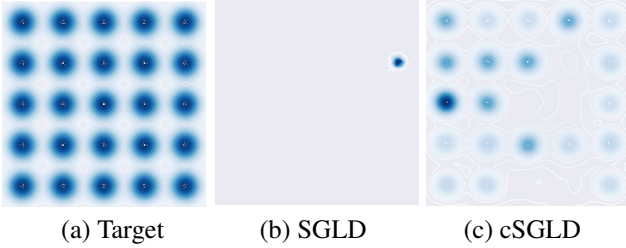In Figure 5, we show the estimated density for SGLD and cSGLD in the non-parallel setting.

(a) Target   (b) SGLD   (c) cSGLD

Figure 5. Sampling from a mixture of 25 Gaussians in the non-parallel setting. With a budget of 50K samples, traditional SGLD has only discovered one of the 25 modes, while our proposed cSGLD has explored significantly more of the distribution.

To quantitatively show the ability of different algorithms to explore multi-modal distributions, we define the *mode-coverage* metric: when the number of samples falling within the radius r of a mode center is larger than a threshold $\bar{n}$, we consider this mode covered. On this dataset, we choose $r = 0.25$ and $n = 100$. Table 4 shows the mode-coverage for several algorithms, based on 10 different runs.

Algorithm	Mode coverage
SGLD	1.8±0.13
cSGLD	**6.7**±0.52
Parallel SGLD	18±0.47
Parallel cSGLD	**24.4**±0.22

Table 4. Mode coverage over 10 different runs, ± standard error.

B. Assumptions

Following Raginsky et al. (2017), we adopt the following standard assumptions summarized in Assumption 2.

Assumption 2.

- There exists some constants $A \geq 0$ and $B \geq 0$, such that $U(0) \leq A$ and $\nabla U(0) \leq B$.

- The function U is L_U-smooth : $\|\nabla U(w) - \nabla U(v)\| \leq L_U \|w - v\|$.

- The function U is $(m_u, b) - dissipative$, which means for some $m_U > 0$ and $b > 0$ $\langle w, \nabla U(w) \rangle \geq m_U \|w\|^2 - b$.

- There exists some constant $\delta \in [0, 1)$, such that $\mathbb{E}[\|\nabla \tilde{U}_k(w) - \nabla U(w)\|^2] \leq 2\sigma(M^2 \|w\|^2 + B^2)$.

- We can choose the μ_0 which satisfies the requirement: $\kappa_0 := \log \int e^{\|w\|^2} \mu_0(w) dw < \infty$.

C. Proof of Theorem 1

Proof of the bound for $W_2(\mu_K, \nu_\infty)$ in cSGLD. Firstly, we introduce the following SDE

$$d\theta_t = -\nabla U(\theta_t) dt + \sqrt{2} d\mathcal{W}_t. \tag{7}$$

Let ν_t denote the distribution of θ_t, and the stationary distribution of (7) be $p(\theta|\mathcal{D})$, which means $\nu_\infty = p(\theta|\mathcal{D})$.

$$\theta_{k+1} = \theta_k - \nabla \tilde{U}_k(\theta_k) \alpha_{k+1} + \sqrt{2\alpha_{k+1}} \xi_{k+1} \tag{8}$$

Further, let μ_k denote the distribution of θ_k.

Since $W_2(\mu_K, \nu_\infty) \leq W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k}) + W_2(\nu_{\sum_{k=1}^K \alpha_k}, \nu_\infty)$, we need to give the bounds for these two parts respectively.

C.1. $W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k})$

For the first part, $W_2(\mu_K, \nu_{\sum_{k=1}^K \alpha_k})$, our proof is based on the proof of Lemma 3.6 in Raginsky et al. (2017) with some modifications. We first assume $\mathbb{E}(\nabla \tilde{U}(w)) = \nabla U(w), \ \forall w \in \mathbb{R}^d$, which is a general assumption according to the way we choose the minibatch. And we define $p(t)$ which will be used in the following proof:

$$p(t) = \{k \in \mathbb{Z}| \sum_{i=1}^k \alpha_i \leq t < \sum_{i=1}^{k+1} \alpha_i\} \tag{9}$$

Then we focus on the following continuous-time interpolation of θ_k:

$$\underline{\theta}(t) = \theta_0 - \int_0^t \nabla\tilde{U}\left(\underline{\theta}(\sum_{k=1}^{p(s)}\alpha_k)\right)\mathrm{d}s + \sqrt{2}\int_0^t d\mathcal{W}_s^{(d)}$$

(10)

where $\nabla\tilde{U} \equiv \nabla\tilde{U}_k$ for $t \in \left[\sum_{i=1}^k \alpha_i, \sum_{i=1}^{k+1}\alpha_i\right)$. And for each k , $\underline{\theta}(\sum_{i=1}^k \alpha_i)$ and θ_k have the same probability law μ_k.

Since $\underline{\theta}(t)$ is not a Markov process, we define the following process which has the same one-time marginals as $\underline{\theta}(t)$

$$V(t) = \theta_0 - \int_0^t G_s\left(V(s)\right)\mathrm{d}s + \sqrt{2}\int_0^t d\mathcal{W}_s^{(d)}$$ (11)

with

$$G_t(x) := \mathbb{E}\left[\nabla\tilde{U}\left(\underline{\theta}(\sum_{i=1}^{q(t)}\alpha_i)\right)|\underline{\theta}(t) = x\right]$$ (12)

Let $\mathbf{P}_V^t := \mathcal{L}\left(V(s) : 0 \leq s \leq t\right)$ and $\mathbf{P}_\theta^t := \mathcal{L}\left(\theta(s) : 0 \leq s \leq t\right)$ and according to the proof of Lemma 3.6 in Raginsky et al. (2017), we can derive a similar result for the relative entropy of $\mathbf{P}_V^t$ and $\mathbf{P}_\theta^t$:

$$D_{KL}(\mathbf{P}_V^t \| \mathbf{P}_\theta^t) = -\int \mathrm{d}\mathbf{P}_V^t \log\frac{\mathrm{d}\mathbf{P}_V^t}{\mathrm{d}\mathbf{P}_\theta^t}$$

$$= \frac{1}{4}\int_0^t \mathbb{E}\|\nabla U(V(s)) - G_s(V(s))\|^2\mathrm{d}s$$

$$= \frac{1}{4}\int_0^t \mathbb{E}\|\nabla U(\underline{\theta}(s)) - G_s(\underline{\theta}(s))\|^2\mathrm{d}s$$

The last line follows the fact that $\mathcal{L}(\underline{\theta}(s)) = \mathcal{L}(V(s))$, $\forall s$. Then we will let $t = \sum_{k=1}^K \alpha_k$ and we can use the martingale property of the integral to derive:

$$D_{KL}(\mathbf{P}_V^{\sum_{k=1}^K \alpha_k} \| \mathbf{P}_\theta^{\sum_{k=1}^K \alpha_k})$$

$$= \frac{1}{4}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(s)) - G_s(\underline{\theta}(s))\|^2\mathrm{d}s$$

$$\leq \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(s)) - \nabla U(\underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i))\|^2\mathrm{d}s$$

$$+ \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i))\|^2\mathrm{d}s$$

$$\leq \frac{L_U^2}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i)\|^2\mathrm{d}s$$ (13)

$$+ \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i))\|^2\mathrm{d}s$$

(14)

For the first part (13), we consider some $s \in [\sum_{k=1}^j \alpha_k, \sum_{k=1}^{j+1}\alpha_k)$, for which the following holds:

$$\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^j \alpha_k) =$$

$$- (s - \sum_{k=1}^j \alpha_k)\nabla\tilde{U}_k(\theta_k) + \sqrt{2}(\mathcal{W}_s^{(d)} - \mathcal{W}_{\sum_{k=1}^j \alpha_k}^{(d)}) =$$

$$- (s - \sum_{k=1}^j \alpha_k)\nabla U(\theta_k) + (s - \sum_{k=1}^j \alpha_k)(\nabla U(\theta_k) - \nabla\tilde{U}_k(\theta_k))$$

$$+ \sqrt{2}(\mathcal{W}_s^{(d)} - \mathcal{W}_{\sum_{k=1}^j \alpha_k}^{(d)})$$ (15)

Thus, we can use Lemma 3.1 and 3.2 in Raginsky et al. (2017) for the following result:

$$\mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^j \alpha_k)\|^2 \leq 3\alpha_{j+1}^2\mathbb{E}\|\nabla U(\theta_j)\|^2$$

$$+ 3\alpha_{j+1}^2\mathbb{E}\|\nabla U(\theta_j) - \nabla\tilde{U}_j(\theta_j)\|^2 + 6\alpha_{j+1}d$$

$$\leq 12\alpha_{j+1}^2(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 6\alpha_{j+1}d$$

Hence we can bound the first part, (choosing $\alpha_0 \leq 1$),

$$\frac{L_U^2}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j \alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^{q(s)}\alpha_i)\|^2\mathrm{d}s$$

$$\leq \frac{L_U^2}{2}\sum_{j=0}^{K-1}\left[12\alpha_{j+1}^3(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 6\alpha_{j+1}^2 d\right]$$

$$\leq L_U^2 \max_{0\leq j\leq K-1}\left[6(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 3d\right](\sum_{j=0}^{K-1}\alpha_{j+1}^2)$$

$$\leq L_U^2 \max_{0\leq j\leq K-1}\left[6(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 3d\right]\frac{3\alpha_0^2 K}{8}$$ (16)

The last line (16) follows from[3]

$$\sum_{j=0}^{K-1}\alpha_{j+1}^2 = \frac{\alpha_0^2}{4}\sum_{j=0}^{K-1}[\cos(\frac{\pi mod(j-1, [K/M])}{[K/M]}) + 1]^2$$

$$= \frac{\alpha_0^2}{4}\sum_{j=0}^{K-1}[\cos^2(\frac{\pi mod(j-1, K/M)}{K/M}) + 1]^2$$

$$= \frac{\alpha_0^2}{4}\frac{K}{M}(\frac{M}{2} + M) = \frac{3\alpha_0^2 K}{8}$$ (17)

---

[3]Note: we only focus on the case when $K \mod M = 0$.

The second part (14) can be bounded as follows:

$$\frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^{j}\alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(\overset{q(s)}{\sum_{k=1}}\alpha_i)) - G_s(\underline{\theta}(\overset{q(s)}{\sum_{k=1}}\alpha_i))\|^2\mathrm{d}s$$

$$=\frac{1}{2}\sum_{j=0}^{K-1}\alpha_{j+1}\mathbb{E}\|\nabla U(\theta_j) - \nabla\tilde{U}(\theta_j)\|^2$$

$$\leq\sigma\max_{0\leq j\leq K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)\sum_{j=0}^{K-1}\alpha_{j+1}$$

$$\leq\sigma\max_{0\leq j\leq K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)(\frac{\alpha_0}{2}\sum_{j=0}^{K-1}(\cos(\frac{\pi mod(j,K/M)}{K/M}) + 1))$$

$$\leq\sigma\max_{0\leq j\leq K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)(\frac{K\alpha_0}{2})$$

Due to the data-processing inequality for the relative entropy, we have

$$D_{KL}(\mu_K\|\nu_{\sum_{k=1}^{K}\alpha_k}) \leq D_{KL}(\mathbf{P}_V^t \| \mathbf{P}_\theta^t)$$

$$\leq\frac{L_U^2}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^{j}\alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\overset{q(s)}{\sum_{k=1}}\alpha_i)\|^2\mathrm{d}s$$

$$+\frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^{j}\alpha_k}^{\sum_{k=1}^{j+1}\alpha_k}\mathbb{E}\|\nabla U(\underline{\theta}(\overset{q(s)}{\sum_{k=1}}\alpha_i)) - G_s(\underline{\theta}(\overset{q(s)}{\sum_{k=1}}\alpha_i))\|^2\mathrm{d}s$$

$$\leq L_U^2\max_{0\leq j\leq K-1}\left[6(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 3d\right]\frac{3\alpha_0^2 K}{8}$$

$$+\sigma\max_{0\leq j\leq K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)(\frac{K\alpha_0}{2})$$

According to the Lemma 3.2 in Raginsky et al. (2017), we have

$$\max_{0\leq j\leq K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)$$
$$\leq (L_U^2(\kappa_0 + 2(1\wedge\frac{1}{m_U})(b + 2B^2 + d)) + B^2) := C_0$$

Then we denote the $6L_U^2(C_0 + d)$ as C_1 and we can derive

$$D_{KL}(\mu_K\|\nu_{\sum_{k=1}^{K}\alpha_k}) \leq C_1(\frac{3\alpha_0^2 K}{8}) + \sigma C_0(\frac{K\alpha_0}{2})$$

Then according to Proposition 3.1 in Bolley & Villani (2005) and Lemma 3.3 in Raginsky et al. (2017), if we denote $\kappa_0 + 2b + 2d$ as C_2, we can derive the following result:

$$W_2(\mu_K, \nu_{\sum_{k=1}^{K}\alpha_k}) \leq$$

$$(12 + C_2(\sum_{k=1}^{K}\alpha_k))^{\frac{1}{2}} \times [D_{KL}(\mu_K\|\nu_{\sum_{k=1}^{K}\alpha_k})^{\frac{1}{2}}$$

$$+ D_{KL}(\mu_K\|\nu_{\sum_{k=1}^{K}\alpha_k})^{\frac{1}{4}}]$$

$$\leq (12 + \frac{C_2 K\alpha_0}{2})^{\frac{1}{2}} \times [(\frac{3C_1\alpha_0^2 K}{8} + \frac{K\sigma C_0\alpha_0}{2})^{\frac{1}{2}}$$

$$+ (\frac{3C_1\alpha_0^2 K}{16} + \frac{K\sigma C_0\alpha_0}{4})^{\frac{1}{4}}]$$

C.2. $W_2(\nu_{\sum_{k=1}^{K}\alpha_k}, \nu_\infty)$

We can directly get the following results from (3.17) in Raginsky et al. (2017) that there exist some positive constants (C_3, C_4),

$$W_2(\nu_{\sum_{k=1}^{K}\alpha_k}, \nu_\infty) \leq C_3\exp(-\sum_{k=1}^{K}\alpha_k/C_4).$$

$\square$

Proof of the bound of $W_2(\tilde{\mu}_K, \nu_\infty)$ in the standard SGLD. Similar to the proof of $W_2(\mu_K, \nu_\infty)$ in cSGLD, we get the following update rule for SGLD with the stepsize following a polynomial decay *i.e.*, $h_k = \frac{h_0}{k}$,

$$\theta_{k+1} = \theta_k - \nabla\tilde{U}_k(\theta_k)h_{k+1} + \sqrt{2h_{k+1}}\xi_{k+1} \qquad (18)$$

Let the $\tilde{\mu}_k$ denote the distribution of θ_k.

Since $W_2(\tilde{\mu}_K, \nu_\infty) \leq W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^{K}h_k}) + W_2(\nu_{\sum_{k=1}^{K}h_k}, \nu_\infty)$, we need to give the bounds for these two parts respectively.

C.3. $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^{K}h_k})$

We first assume $\mathbb{E}(\nabla\tilde{U}(w)) = \nabla U(w), \ \forall w\in\mathbb{R}^d$, which is a general assumption according to the way we choose the minibatch. Following the proof in Raginsky et al. (2017) and the analysis of the SPOS method in Zhang et al., we define the following $p(t)$ which will be used in the following proof:

$$p(t) = \{k\in\mathbb{Z}|\sum_{i=1}^{k}h_i \leq t < \sum_{i=1}^{k+1}h_i\} \qquad (19)$$

Then we focus on the following continuous-time interpolation of θ_k:

$$\underline{\theta}(t) = \theta_0 - \int_0^t \nabla\tilde{U}\left(\underline{\theta}(\overset{p(s)}{\sum_{k=1}}h_k)\right)\mathrm{d}s + \sqrt{2}\int_0^t d\mathcal{W}_s^{(d)},$$

$$(20)$$
$$(21)$$

where $\nabla\tilde{U} \equiv \nabla\tilde{U}_k$ for $t\in\left[\sum_{i=1}^{k}h_i, \sum_{i=1}^{k+1}h_i\right)$. And for each k, $\underline{\theta}(\sum_{i=1}^{k}h_i)$ and θ_k have the same probability law $\tilde{\mu}_k$.

Since $\underline{\theta}(t)$ is not a Markov process, we define the following process which has the same one-time marginals as $\underline{\theta}(t)$

$$V(t) = \theta_0 - \int_0^t G_s(V(s))\mathrm{d}s + \sqrt{2}\int_0^t d\mathcal{W}_s^{(d)} \qquad (22)$$

with

$$G_t(x) := \mathbb{E}\left[\nabla\tilde{U}\left(\underline{\theta}(\sum_{i=1}^{q(t)} h_i)\right) \Big| \underline{\theta}(t) = x \right] \qquad (23)$$

Let $\mathbf{P}_V^t := \mathcal{L}(V(s) : 0 \le s \le t)$ and $\mathbf{P}_\theta^t := \mathcal{L}(\theta(s) : 0 \le s \le t)$ and according to the proof of Lemma 3.6 in Raginsky et al. (2017), we can derive the similar result for the relative entropy of $\mathbf{P}_V^t$ and $\mathbf{P}_\theta^t$:

$$\begin{aligned}
D_{KL}(\mathbf{P}_V^t \,\|\, \mathbf{P}_\theta^t) &= -\int \mathrm{d}\mathbf{P}_V^t \log \frac{\mathrm{d}\mathbf{P}_V^t}{\mathrm{d}\mathbf{P}_\theta^t} \\
&= \frac{1}{4}\int_0^t \mathbb{E}\|\nabla U(V(s)) - G_s(V(s))\|^2 \mathrm{d}s \\
&= \frac{1}{4}\int_0^t \mathbb{E}\|\nabla U(\underline{\theta}(s)) - G_s(\underline{\theta}(s))\|^2 \mathrm{d}s
\end{aligned}$$

The last line follows the fact that $\mathcal{L}(\underline{\theta}(s)) = \mathcal{L}(V(s))$, $\forall s$. Then we will let $t = \sum_{k=1}^K h_k$ and we can use the martingale property of integral to derive:

$$\begin{aligned}
&D_{KL}(\mathbf{P}_V^{\sum_{k=1}^K h_k} \,\|\, \mathbf{P}_\theta^{\sum_{k=1}^K h_k}) \\
&= \frac{1}{4}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(s)) - G_s(\underline{\theta}(s))\|^2 \mathrm{d}s \\
&\le \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(s)) - \nabla U(\underline{\theta}(\sum_{k=1}^{q(s)} h_i))\|^2 \mathrm{d}s \\
&+ \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)} h_i)) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)} h_i))\|^2 \mathrm{d}s \\
&\le \frac{L_U^2}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^{q(s)} h_i)\|^2 \mathrm{d}s \qquad (24)
\end{aligned}$$

$$+ \frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)} h_i)) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)} h_i))\|^2 \mathrm{d}s$$
$$(25)$$

For the first part (24), we consider some $s \in [\sum_{k=1}^j h_k, \sum_{k=1}^{j+1} h_k)$, where the following equation holds:

$$\begin{aligned}
&\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^j h_k) = \\
&- (s - \sum_{k=1}^j h_k)\nabla\tilde{U}_k(\theta_k) + \sqrt{2}(\mathcal{W}_s^{(d)} - \mathcal{W}_{\sum_{k=1}^j h_k}^{(d)}) = \\
&- (s - \sum_{k=1}^j h_k)\nabla U(\theta_k) + (s - \sum_{k=1}^j h_k)(\nabla U(\theta_k) - \nabla\tilde{U}_k(\theta_k)) \\
&+ \sqrt{2}(\mathcal{W}_s^{(d)} - \mathcal{W}_{\sum_{k=1}^j h_k}^{(d)}) \qquad (26)
\end{aligned}$$

Thus, we can use Lemma 3.1 and 3.2 in Raginsky et al. (2017) for the following result:

$$\begin{aligned}
\mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^j h_k)\|^2 &\le 3h_{j+1}^2\mathbb{E}\|\nabla U(\theta_j)\|^2 \\
&+ 3h_{j+1}^2\mathbb{E}\|\nabla U(\theta_j) - \nabla\tilde{U}_j(\theta_j)\|^2 + 6h_{j+1}d \\
&\le 12h_{j+1}^2(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 6h_{j+1}d
\end{aligned}$$

Hence we can bound the first part, (choosing $h_0 \le 1$),

$$\begin{aligned}
&\frac{L_U^2}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^{q(s)} h_k)\|^2 \mathrm{d}s \\
&\le \frac{L_U^2}{2}\sum_{j=0}^{K-1}\left[12h_{j+1}^3(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 6h_{j+1}^2 d\right] \\
&\le L_U^2 \max_{0\le j\le K-1}\left[6(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 3d\right](\sum_{j=0}^{K-1} h_{j+1}^2) \\
&\le L_U^2 \max_{0\le j\le K-1}\left[6(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2) + 3d\right]\frac{\pi^2}{6}h_0^2 \qquad (27)
\end{aligned}$$

where the last line follows from the fact that

$$\sum_{j=0}^{K-1}\frac{1}{(j+1)^3} \le \sum_{j=0}^{K-1}\frac{1}{(j+1)^2} \le \sum_{j=0}^{\infty}\frac{1}{(j+1)^2} = \frac{\pi^2}{6}.$$

The second part (25) can be bounded as follows:

$$\begin{aligned}
&\frac{1}{2}\sum_{j=0}^{K-1}\int_{\sum_{k=1}^j h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)} h_i)) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)} h_i))\|^2 \mathrm{d}s \\
&= \frac{1}{2}\sum_{j=0}^{K-1} h_{j+1}\mathbb{E}\|\nabla U(\theta_j) - \nabla\tilde{U}(\theta_j)\|^2 \\
&\le \sigma \max_{0\le j\le K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)\sum_{j=0}^{K-1} h_{j+1} \\
&\le \sigma \max_{0\le j\le K-1}(L_U^2\mathbb{E}\|\theta_j\|^2 + B^2)(h_0\sum_{j=1}^K \frac{1}{j})
\end{aligned}$$

Due to the data-processing inequality for the relative en-

tropy, we have

$$D_{KL}(\tilde{\mu}_K \| \nu_{\sum_{k=1}^{K} h_k}) \leq D_{KL}(\mathbf{P}_V^t \| \mathbf{P}_\theta^t)$$

$$\leq \frac{L_U^2}{2} \sum_{j=0}^{K-1} \int_{\sum_{k=1}^{j} h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\underline{\theta}(s) - \underline{\theta}(\sum_{k=1}^{q(s)} h_i)\|^2 \mathrm{d}s$$

$$+ \frac{1}{2} \sum_{j=0}^{K-1} \int_{\sum_{k=1}^{j} h_k}^{\sum_{k=1}^{j+1} h_k} \mathbb{E}\|\nabla U(\underline{\theta}(\sum_{k=1}^{q(s)} h_i)) - G_s(\underline{\theta}(\sum_{k=1}^{q(s)} h_i)\|^2 \mathrm{d}s$$

$$\leq L_U^2 \max_{0 \leq j \leq K-1} \left[6(L_U^2 \mathbb{E}\|\theta_j\|^2 + B^2) + 3d \right] \frac{\pi^2}{6} h_0^2$$

$$+ \sigma \max_{0 \leq j \leq K-1} (L_U^2 \mathbb{E}\|\theta_j\|^2 + B^2)(h_0 \sum_{j=1}^{K} \frac{1}{j})$$

According to Lemma 3.2 in Raginsky et al. (2017), we have

$$\max_{0 \leq j \leq K-1} (L_U^2 \mathbb{E}\|\theta_j\|^2 + B^2)$$

$$\leq (L_U^2(\kappa_0 + 2(1 \wedge \frac{1}{m_U})(b + 2B^2 + d)) + B^2) := C_0$$

Then we denote the $6L_U^2(C_0 + d)$ as C_1 and we can derive

$$D_{KL}(\tilde{\mu}_K \| \nu_{\sum_{k=1}^{K} h_k}) \leq C_1 h_0^2 \frac{\pi^2}{6} + \sigma C_0 h_0 \sum_{j=1}^{K} \frac{1}{j}$$

Then according to Proposition 3.1 in Bolley & Villani (2005) and Lemma 3.3 in Raginsky et al. (2017), if we denote $\kappa_0 + 2b + 2d$ as C_2, we can derive the following result,

$$W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^{K} h_k}) \leq$$

$$[12 + C_2(\sum_{k=1}^{K} h_k)]^{1/2} \times [(D_{KL}(\tilde{\mu}_K \| \nu_{\sum_{k=1}^{K} h_k}))^{1/2} +$$

$$(D_{KL}(\tilde{\mu}_K \| \nu_{\sum_{k=1}^{K} h_k})/2)^{1/4}]$$

$$= [12 + C_2(h_0 \sum_{j=1}^{K} \frac{1}{j})]^{1/2} \times [(C_1 h_0^2 \frac{\pi^2}{6} + \sigma C_0 h_0 \sum_{j=1}^{K} \frac{1}{j})^{1/2} +$$

$$(C_1 h_0^2 \frac{\pi^2}{12} + \sigma C_0 h_0 \sum_{j=1}^{K} \frac{1}{2j})^{1/4}]$$

Now we derive the bound for $W_2(\tilde{\mu}_K, \nu_{\sum_{k=1}^{K} h_k})$.

C.4. $W_2(\nu_{\sum_{k=1}^{K} h_k}, \nu_\infty)$

We can directly get the following results from (3.17) in (Raginsky et al., 2017) that there exist some positive constants (C_3, C_4),

$$W_2(\nu_{\sum_{k=1}^{K} h_k}, \nu_\infty) \leq C_3 \exp(-\sum_{k=1}^{K} h_k/C_4)$$

$\square$