Low-Precision Stochastic Gradient Langevin Dynamics

Ruqi Zhang [1] **Andrew Gordon Wilson** [2] **Christopher De Sa** [3]

Abstract

While low-precision optimization has been widely used to accelerate deep learning, low-precision sampling remains largely unexplored. As a consequence, sampling is simply infeasible in many large-scale scenarios, despite providing remarkable benefits to generalization and uncertainty estimation for neural networks. In this paper, we provide the first study of low-precision Stochastic Gradient Langevin Dynamics (SGLD), showing that its costs can be significantly reduced without sacrificing performance, due to its intrinsic ability to handle system noise. We prove that the convergence of low-precision SGLD with full-precision gradient accumulators is less affected by the quantization error than its SGD counterpart in the strongly convex setting. To further enable low-precision gradient accumulators, we develop a new quantization function for SGLD that preserves the variance in each update step. We demonstrate that low-precision SGLD achieves comparable performance to full-precision SGLD with only 8 bits on a variety of deep learning tasks.

1. Introduction

Low-precision optimization has become increasingly popular in reducing computation and memory costs of training deep neural networks (DNNs). It uses fewer bits to represent numbers in model parameters, activations, and gradients, and thus can drastically lower resource demands (Gupta et al., 2015; Zhou et al., 2016; De Sa et al., 2017; Li et al., 2017). Prior work has shown that using 8-bit numbers in training DNNs achieves about 4× latency speed ups and memory reduction compared to 32-bit numbers on a wide variety of deep learning tasks (Sun et al., 2019; Yang et al., 2019; Wang et al., 2018b; Banner et al., 2018). As datasets and architectures grow rapidly, performing low-precision optimization enables training large-scale DNNs efficiently and enables many applications on different hardware and platforms.

Despite the impressive progress in low-precision optimization, low-precision sampling remains largely unexplored. However, we believe stochastic gradient Markov chain Monte Carlo (SGMCMC) methods (Welling & Teh, 2011; Chen et al., 2014; Ma et al., 2015) are particularly suited for low-precision arithmetic because of their intrinsic robustness to system noise. In particular: (1) SGMCMC explores weight space instead of converging to a single point, thus it should not require precise weights or gradients; (2) SGMCMC even adds noise to the system to encourage exploration and so is naturally more tolerant to quantization noise; (3) SGMCMC performs Bayesian model averaging during testing using an ensemble of models, which allows coarse representations of individual models to be compensated by the overall model average (Zhu et al., 2019).

SGMCMC is particularly compelling in Bayesian deep learning due to its ability to characterize complex and multi-modal DNN posteriors, providing state-of-the-art generalization accuracy and calibration (Zhang et al., 2020; Li et al., 2016; Gan et al., 2017; Heek & Kalchbrenner, 2019). Moreover, low-precision approaches are especially appealing in this setting, where at test time we must store samples from a posterior over millions of parameters, and perform multiple forward passes through the corresponding models, which incurs significant memory and computational expenses.

In this paper, we give the first comprehensive study of low-precision Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011), providing both theoretical convergence bounds and promising empirical results in deep learning. On strongly log-concave distributions (i.e. strongly convex functions for SGD), we prove that the convergence of SGLD with full-precision gradient accumulators is more robust to the quantization error than its counterpart in SGD. Surprisingly, we find that SGLD with low-precision gradient accumulators can diverge arbitrarily far away from the target distribution with small stepsizes. We identify the source of the issue and develop a new quantization function to correct the bias with minimal overhead. Empirically, we demonstrate low-precision SGLD across different tasks, showing that it is able to provide superior

[1]The University of Texas at Austin [2]New York University [3]Cornell University. Correspondence to: Ruqi Zhang <ruqiz@utexas.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

generalization and uncertainty estimation using just 8 bits. We summarize our contributions as follows:

- We provide a methodology for SGLD to leverage low-precision computation, including a new quantization function, while still guaranteeing its convergence to the target distribution.
- We offer theoretical results which explicitly show how quantization error affects the convergence of the sampler in the strongly convex setting, proving its robustness to quantization noise over SGD.
- We show SGLD is particularly suitable for low-precision deep learning over a range of experiments, including logistic regression, Bayesian neural networks on image and text classification.

In short, low-precision SGLD is often a compelling alternative to standard SGLD, improving speed and memory efficiency, while retaining accuracy. Moreover, SGLD is arguably more amenable to low-precision computations than SGD. Our code is available here.

2. Related Work

To speed up SGLD training, most existing work is on distributed learning with synchronous or asynchronous communication (Ahn et al., 2014; Chen et al., 2016; Li et al., 2019). Another direction is to shorten training time by accelerating the convergence using variance reduction techniques (Dubey et al., 2016; Baker et al., 2019), importance sampling (Deng et al., 2020) or a cyclical learning rate schedule (Zhang et al., 2020). To speed up SGLD during testing, distillation techniques are often used to save both compute and memory which transfer the knowledge of an ensemble of models to a single model (Korattikara et al., 2015; Wang et al., 2018a).

Low-precision computation has become one of the most common approaches to reduce latency and memory consumption in deep learning and is widely supported on new emerging chips including CPUs, GPUs and TPUs (Micikevicius et al., 2018; Krishnamoorthi, 2018; Esser et al., 2020). Two main directions to improve low-precision training include developing new number formats (Sun et al., 2019; 2020) or studying mixed-precision schemes (Courbariaux et al., 2015; Zhou et al., 2016; Banner et al., 2018). Recently, one line of work applies the Bayesian framework to learn a deterministic quantized neural network (Soudry et al., 2014; Cheng et al., 2015; Achterhold et al., 2018; van Baalen et al., 2020; Meng et al., 2020).

Low-precision computation is largely unexplored for Bayesian neural networks, despite their specific promise in this domain. Su et al. (2019) proposes a method to train binarized variational BNNs, and Cai et al. (2018) develops efficient hardware for training low-precision variational BNNs. The only work on low-precision MCMC known to us is Ferianc et al. (2021), which directly applies *post-training* quantization techniques from optimization (Jacob et al., 2018) to convert BNNs trained by Stochastic Gradient Hamiltonian Monte Carlo (Chen et al., 2014) into low-precision models. We instead study training low-precision models by SGLD from scratch, to accelerate both training and testing.

3. Preliminaries

3.1. Stochastic Gradient Langevin Dynamics

In the Bayesian setting, given some dataset $\mathcal{D}$, a model with parameters $\theta \in \mathbb{R}^d$, and a prior $p(\theta)$, we are interested in sampling from the posterior $p(\theta|\mathcal{D}) \propto \exp(-U(\theta))$, where the energy function is

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

When the dataset is large, the cost of computing a sum over the entire dataset is expensive. Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011) reduces the cost by using a stochastic gradient estimation $\nabla\tilde{U}$, which is an unbiased estimator of ∇U based on a subset of the dataset $\mathcal{D}$. Specifically, SGLD updates the parameter θ in the $(k+1)$-th step following the rule

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

where α is the stepsize and ξ is a standard Gaussian noise. Compared to the SGD update, the only difference is that SGLD adds an additional Gaussian noise in each step, which essentially enables SGLD to characterize the full distribution instead of converging to a single point. The close connection between SGLD and SGD makes it convenient to implement and run on existing deep learning tasks for which SGD is the typical learning algorithm.

3.2. Low-Precision Training

We study training a low-precision model by SGLD from scratch, to reduce both training and testing costs. Specifically, we follow the framework in prior work to quantize the weights, activations, backpropagation errors, and gradients (Wu et al., 2018; Wang et al., 2018b; Yang et al., 2019). We mainly consider the effect of weight and gradient quantization following previous work (Li et al., 2017; Yang et al., 2019). Please refer to Appendix A for more details.

3.2.1. Number Representations

To represent numbers in low-precision, one simple way is to use *fixed point*, which has been utilized in both theory and

practice (Gupta et al., 2015; Lin et al., 2016; Li et al., 2017; Yang et al., 2019). Specifically, suppose that we use W bits to represent numbers with F of those W bits to represent the fractional part. Then there is a distance between consecutive representable numbers, $\Delta = 2^{-F}$, which is called *quantization gap*. The representable numbers also have a lower bound l and an upper bound u, where

$$l = -2^{W-F-1}, \quad u = 2^{W-F-1} - 2^{-F}.$$

As shown above, when the number of bits decreases, the accuracy of number representation decreases. We use this type of number representation in our theoretical analysis and empirical demonstration following previous work (Li et al., 2017; Yang et al., 2019).

Another common type of number representation is *floating point* where each number has its own exponent part. Between fixed point and floating point, there is *block floating point* which allows all numbers within a block to share the same exponent (Song et al., 2018). We use block floating point for deep learning experiments since it has been shown more favorable for deep models (Yang et al., 2019).

3.2.2. Quantization

Having low-precision number representation in hand, we also need a quantization function Q to convert a real-valued number into a low-precision number. Such functions include *deterministic rounding* and *stochastic rounding*. Particularly, the deterministic rounding function Q^d quantizes a number to its nearest representable neighbor as follows:

$$Q^d(\theta) = \text{sign}(\theta) \cdot \text{clip}\left(\Delta \left\lfloor \frac{|\theta|}{\Delta} + \frac{1}{2} \right\rfloor, l, u\right),$$

where $\text{clip}(x, l, u) = \max(\min(x, u), l)$. Instead, stochastic rounding Q^s quantizes a number with a probability based on the distance to its representable neighbor:

$$Q^s(\theta) = \begin{cases} \text{clip}\left(\Delta \left\lfloor \frac{\theta}{\Delta} \right\rfloor, l, u\right), & \text{w.p. } \left\lceil \frac{\theta}{\Delta} \right\rceil - \frac{\theta}{\Delta} \\ \text{clip}\left(\Delta \left\lceil \frac{\theta}{\Delta} \right\rceil, l, u\right), & \text{w.p. } 1 - \left(\left\lceil \frac{\theta}{\Delta} \right\rceil - \frac{\theta}{\Delta}\right). \end{cases}$$

An important property of Q^s is that $\mathbf{E}[Q^s(\theta)] = \theta$, which means the quantized number is unbiased. Q^s is generally preferred over Q^d in practice since it can preserve gradient information especially when the gradient update is smaller than the quantization gap (Gupta et al., 2015; Wang et al., 2018b). In what follows, we use Q_W and Δ_W to denote the weights' quantizer and quantization gap, Q_G and Δ_G to denote gradients' quantizer and quantization gap.

To do the gradient update in low-precision training, there are two common choices depending on whether we store an additional copy of full-precision weights. *Full-precision gradient accumulators* use a full-precision weight buffer to accumulate gradient updates and only quantize weights before computing gradients. SGD with full-precision gradient accumulators (SGDLP-F) updates the weights as the following,

$$\theta_{k+1} = \theta_k - \alpha Q_G\left(\nabla \tilde{U}(Q_W(\theta_k))\right),$$

where we use full-precision θ_{k+1} and θ_k in the update, and only quantize the weight for forward and backward propagation (Courbariaux et al., 2015; Li et al., 2017).

However, gradient accumulators have to be frequently updated during training, therefore it will be ideal to also represent it in low-precision to further reduce the costs. To achieve it, we could instead do the update as follows,

$$\theta_{k+1} = Q_W\left(\theta_k - \alpha Q_G\left(\nabla \tilde{U}(\theta_k)\right)\right), \tag{2}$$

where θ is always represented in low-precision. This update of SGD is called using *low-precision gradient accumulators* (SGDLP-L). Both full- and low-precision gradient accumulators have been widely used: low-precision gradient accumulators are cheaper and faster because of having all numbers in low-precision, whereas full-precision gradient accumulators generally have better performance because of more precisely reflecting small gradient updates (Courbariaux et al., 2015; Li et al., 2017).

4. Low-Precision SGLD

In this section, we first study the convergence of low-precision SGLD with full-precision gradient accumulators (SGLDLP-F) on strongly log-concave distributions (i.e. the energy function is strongly convex) and show that SGLDLP-F is less affected by the quantization error than its SGD counterpart. Next we analyze low-precision SGLD with low-precision gradient accumulators (SGLDLP-L) under the same setup and prove that SGLDLP-L can diverge arbitrarily far away from the target distribution with a small stepsize, which however is typically required by SGLD to reduce asymptotic bias. Finally, we solve this problem by developing a variance-corrected quantization function and further prove that with this quantization function, SGLDLP-L converges with small stepsizes.

4.1. Full-Precision Gradient Accumulators

As shown in Equation (1), the update of SGLD is simply a SGD update plus a Gaussian noise. Therefore the low-precision formulation for SGD in Section 3.2 can be naturally extended to SGLD training. Similar to SGDLP-F, we can do low-precision SGLD with full-precision gradient accumulators (SGLDLP-F) as the following:

$$\theta_{k+1} = \theta_k - \alpha Q_G\left(\nabla \tilde{U}(Q_W(\theta_k))\right) + \sqrt{2\alpha}\xi_{k+1}, \tag{3}$$

which can also be viewed as SGDLP-F plus a Gaussian noise in each step. However, the Gaussian noise turns out to help counter-effect the noise introduced by quantization, making SGLDLP-F more robust to inaccurate number representation and converging better than SGDLP-F as we show later in this section.

We now prove the convergence of SGLDLP-F. Our analysis is built upon the 2-Wasserstein distance bounds of SGLD in Dalalyan & Karagulyan (2019), where the target distribution is assumed to be smooth and strongly log-concave. We additionally assume the energy function has Lipschitz Hessian following recent work in low-precision optimization (Yang et al., 2019). In summary, the energy function U has the following assumptions, $\forall\, \theta, \theta' \in \mathbb{R}^d$, it satisfies

$$\begin{cases} U(\theta) - U(\theta') - \nabla U(\theta')^\intercal(\theta - \theta') \geq (m/2)\left\|\theta - \theta'\right\|_2^2, \\ \left\|\nabla U(\theta) - \nabla U(\theta')\right\|_2 \leq M\left\|\theta - \theta'\right\|_2, \\ \left\|\nabla^2 U(\theta) - \nabla^2 U(\theta')\right\|_2 \leq \Psi\|\theta - \theta'\|_2, \end{cases}$$

for some positive constants m, M and Ψ. We further assume that the variance of the stochastic gradient is bounded $\mathbf{E}[\|\nabla\tilde{U}(\theta) - \nabla U(\theta)\|_2^2] \leq \kappa^2$ for some constant κ. For simplicity, we consider SGLD with a constant stepsize α. We use stochastic rounding for quantizing both weights and gradients as it is generally better than deterministic rounding and has also been used in previous low-precision theoretical analysis (Li et al., 2017; Yang et al., 2019).

Theorem 1. *We run SGLDLP-F under the above assumptions and with a constant stepsize $\alpha \leq 2/(m+M)$. Let π be the target distribution, μ_0 be the initial distribution and μ_K be the distribution obtained by SGLDLP-F after K iterations, then the 2-Wasserstein distance is*

$$W_2(\mu_K, \pi) \leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) + \sqrt{\frac{(\Delta_G^2 + M^2\Delta_W^2)\alpha d + 4\alpha\kappa^2}{4m}}.$$

This theorem shows that SGLDLP-F converges to the accuracy floor $\min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right)$ given large enough number of iterations K and small enough stepsize α. Besides, if we further assume that the energy function is quadratic, that is $\Psi = 0$, then SGLDLP-F converges to the target distribution asymptotically. This matches the result of SGDLP-F on a quadratic function which converges to the optimum asymptotically (Li et al., 2017). Our theorem also recovers the bound in Dalalyan & Karagulyan (2019) when the quantization gap is zero (we ignore $1.65M$ in the denominator in their bound for simplicity).

However, when the energy function is not quadratic, the convergence of SGLDLP-F to the target distribution has a $\mathcal{O}(\Delta_W^2)$ rate whereas SGDLP-F to the optimum has a $\mathcal{O}(\Delta_W)$ rate (Yang et al., 2019)[1]. Recall that $\Delta_W = 2^{-F}$ where F is the number of fractional bits, our result suggests that asymptotically, SGLD only needs half the number of bits as SGD needs to achieve the same convergence accuracy! Our comparison between SGLD and SGD also fits into the literature in comparing sampling and optimization convergence bounds (Ma et al., 2019; Talwar, 2019) (see Appendix E for more details). In summary, our theorem implies how sensitive SGLD is to the quantization error, and actually suggests that sampling methods are more suitable with low-precision computation than optimization methods.

4.2. Low-Precision Gradient Accumulators

As mentioned before, it will be ideal to further reduce the costs using low-precision gradient accumulators. Mimicking the update of SGDLP-L in Equation (2), it is natural to get the following update rule for SGLD with low-precision gradient accumulators (SGLDLP-L),

$$\theta_{k+1} = Q_W\left(\theta_k - \alpha Q_G\left(\nabla\tilde{U}(\theta_k)\right) + \sqrt{2\alpha}\xi_{k+1}\right). \quad (4)$$

Surprisingly, while we can prove a convergence result for SGLDLP-L, our theory and empirical results suggest that it can diverge arbitrarily far away from the target distribution with small stepsizes.

Theorem 2. *We run SGLDLP-L under the same assumptions as in Theorem 1. Let μ_0 be the initial distribution and μ_K be the distribution obtained by SGLDLP-L after K iterations, then*

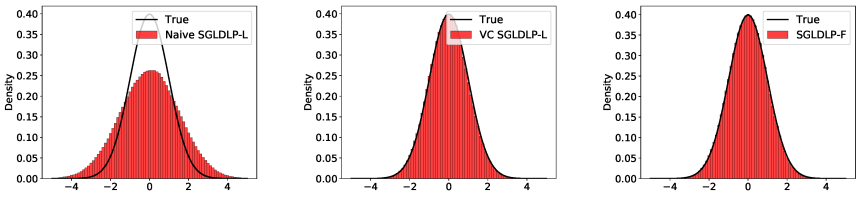
$$W_2(\mu_K, \pi) \leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) + A + \left((1-\alpha m)^K + 1\right)\frac{\Delta_W\sqrt{d}}{2},$$

where $A = \sqrt{\frac{(\alpha\Delta_G^2 + \alpha^{-1}\Delta_W^2)d + 4\alpha\kappa^2}{4m}}$.

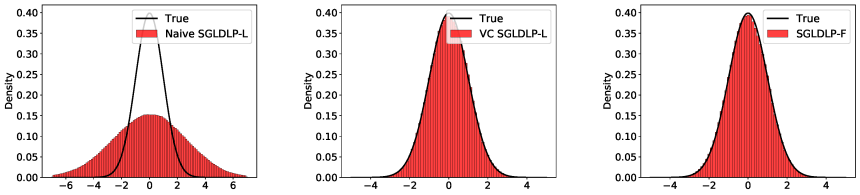
Since the term A contains α^{-1} in the numerator, this theorem implies that as the stepsize α decreases, W_2 distance between the stationary distribution of SGLDLP-L and the target distribution may increase. To test if this is the case, we empirically run SGLDLP-L on a standard Gaussian distribution in Figure 1. We use 8-bit fixed point and assign 3 of them to represent the fractional part. Our results verify that SGLDLP-L indeed diverges from the target distribution with small stepsizes. In the same time, SGLDLP-F always converges to the target distribution with different stepsizes, aligning with the result in Theorem 1.

One may choose a stepsize that minimizes the above W_2 distance to avoid divergence, however, getting that optimal

[1] Their bound $\mathcal{O}(\Delta_W^2)$ is stated for the squared norm therefore we take its square root to compare with our W_2 distance bound which is stated for the norm.



(a) SGLD with $\alpha = 0.001$



(b) SGLD with $\alpha = 0.0001$

Figure 1. Low-precision SGLD with varying stepsizes on a Gaussian distribution. Variance-corrected SGLD with low-precision gradient accumulators (VC SGLDLP-L) and SGLD with full-precision gradient accumulators (SGLDLP-F) converge to the true distribution, whereas naïve SGLDLP-L diverges and the divergence increases as the stepsize decreases.

stepsize is in general difficult since the constants are unknown in practice. Moreover, enabling a small stepsize in SGLD is often desirable, since it is needed to reduce the asymptotic bias of the posterior approximation (Welling & Teh, 2011).

4.3. Variance-Corrected Quantization

To approach correcting the problem with the naïve SGLD with low-precision gradient accumulators, we first need to identify the source of the issue. We show in the following that the reason is the *variance* of each dimension of θ_{k+1} becomes larger due to using low-precision gradient accumulators. Specifically, given the stochastic gradient $\nabla\tilde{U}$, the update of full-precision SGLD can be viewed as sampling from a Gaussian distribution for each dimension i

$$\theta_{k+1,i} \sim \mathcal{N}\left(\theta_{k,i} - \alpha\nabla\tilde{U}(\theta_k)_i, 2\alpha\right), \text{for } i = 1, \cdots, d.$$

Since stochastic rounding is unbiased, using it as the weight quantizer Q_W and the gradient quantizer Q_G in SGLDLP-L gives us

$$\begin{aligned}\mathbf{E}\left[\theta_{k+1,i}\right] &= \mathbf{E}\left[Q_W\left(\theta_{k,i} - \alpha Q_G\left(\nabla\tilde{U}(\theta_k)\right)_i + \sqrt{2\alpha}\xi_{k+1,i}\right)\right] \\ &= \theta_{k,i} - \alpha\nabla\tilde{U}(\theta_k)_i,\end{aligned}$$

which has the same mean as θ_{k+1} in full-precision. However, the variance of $\theta_{k+1,i}$ is now essentially larger than needed. If we ignore the variance from Q_G and the stochastic gradient, since they are present and have been shown to work well in SGLDLP-F, the variance of $\theta_{k+1,i}$ is

$$\begin{aligned}&\mathrm{Var}\left[\theta_{k+1,i}\right] \\ &= \mathbf{E}\left[\mathrm{Var}\left[Q_W\left(\theta_{k,i} - \alpha\nabla U(\theta_k)_i + \sqrt{2\alpha}\xi_{k+1,i}\right)\middle|\xi_{k+1,i}\right]\right] \\ &+ \mathrm{Var}\left[\mathbf{E}\left[Q_W\left(\theta_{k,i} - \alpha\nabla U(\theta_k)_i + \sqrt{2\alpha}\xi_{k+1,i}\right)\middle|\xi_{k+1,i}\right]\right] \\ &= \frac{\Delta_W^2}{4}\chi_{k+1,i} + 2\alpha.\end{aligned}$$

where $\chi_{k+1,i} \in [0, 1]$ depends on the distance of $\theta_{k+1,i}$ to its discrete neighbor. The above equation shows that the variance in SGLDLP-L update is larger than the right variance value 2α. Empirically, from Figure 1, we can also find that naïve SGLDLP-L estimates the mean correctly but variance wrongly. This validates our intuition that low-precision gradient accumulators with stochastic rounding adds more variance than needed leading to an inaccurate variance estimation. Besides, we cannot simply use deterministic rounding to solve the problem, since it is a biased estimation and generally provides much worse results than stochastic rounding especially on deep neural networks (see Appendix F for an empirical demonstration).

To enable SGLD with low-precision gradient accumulators, we propose a new quantization function Q^{vc}, denoting variance-corrected quantization, to fix the issue. The main idea of Q^{vc} is to directly sample from the discrete weight space instead of quantizing a real-valued Gaussian sample. First we note that if we want a sample with mean $\mu \geq 0$ and variance $v \leq \Delta_W^2/4$, we could sample from the following

Algorithm 1 Variance-Corrected Low-Precision SGLD (VC SGLDLP-L).

given: Stepsize α, number of training iterations K, gradient quantizer Q_G and quantization gap of weights Δ_W.
for $k = 1 : K$ **do**
  update $\theta_{k+1} \leftarrow Q^{vc}\left(\theta_k - \alpha Q_G\left(\nabla\tilde{U}(\theta_k)\right), 2\alpha, \Delta_W\right)$
end for
output: samples $\{\theta_k\}$

categorical distribution over $\{\Delta_W, -\Delta_W, 0\}$ to get it,

$$\text{Cat}(\mu, v) = \begin{cases} \Delta_W, & w.p. \frac{v+\mu^2+\mu\Delta_W}{2\Delta_W^2} \\ -\Delta_W, & w.p. \frac{v+\mu^2-\mu\Delta_W}{2\Delta_W^2} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Now we show how to use this categorical distribution to preserve the correct mean and variance for quantized θ_{k+1}. We do so considering two cases: when the Gaussian variance 2α is larger than the largest possible stochastic rounding variance $\Delta_W^2/4$, Q^{vc} first adds a small Gaussian noise and uses a sample from Equation (5) to make up the remaining variance; in the other situation, Q^{vc} directly samples from Equation (5) to achieve the target variance. The full description of Q^{vc} is outlined in Algorithm 2.

Our variance-corrected quantization function Q^{vc} always guarantees the correct mean, $\mathbf{E}\left[\theta_{k+1,i}\right] = \theta_{k,i} - \alpha\nabla\tilde{U}(\theta_k)_i$, and further guarantees the correct variance $\text{Var}\left[\theta_{k+1,i}\right] = 2\alpha$ most of the time except when $v = 2\alpha < v_s$. However that case rarely happens in practice, because the stepsize has to be extremely small. Besides, our quantization is simple to implement and its cost is negligible compared to gradient computation. Although Q^{vc} only preserves the correctness of the first two moments (i.e. mean and variance), we show that this does not affect the performance much in both theory and practice.

We now prove that SGLDLP-L using Q^{vc}, denoting VC SGLDLP-L, converges to the target distribution up to a certain accuracy level with small stepsizes.

Theorem 3. *We run VC SGLDLP-L as in Algorithm 1. Besides the same assumptions in Theorem 1, we further assume the gradient is bounded* $\mathbf{E}\left[\left\|Q_G(\nabla\tilde{U}(\theta_k))\right\|_1\right] \leq G$. *Let* $v_0 = \Delta_W^2/4$. *Then*

$$W_2(\mu_K, \pi) \leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi A}{m}, \frac{M\sqrt{A}}{m}\right) + \sqrt{\frac{\alpha\Delta_G^2 d + 4\alpha\kappa^2}{4m} + \frac{A}{\alpha m}} + \left((1-\alpha m)^K + 1\right)\sqrt{A},$$

where $A = \begin{cases} 5v_0 d, & \text{if } 2\alpha > v_0 \\ \max\left(2\Delta_W\alpha G, 4\alpha d\right), & \text{otherwise} \end{cases}$

Algorithm 2 Variance-Corrected Quantization Function Q^{vc}.

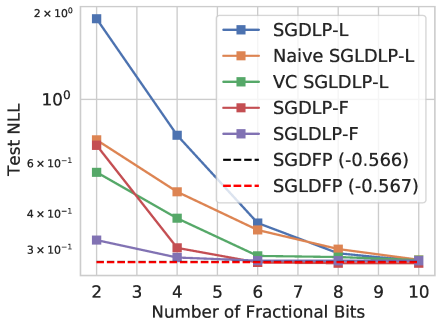
input: (μ, v, Δ) {Q^{vc} returns a variable with mean μ and variance v}
$v_0 \leftarrow \Delta^2/4$ {$\Delta^2/4$ is the largest possible variance that stochastic rounding can cause}
if $v > v_0$ **then** {add a small Gaussian noise and sample from the discrete grid to make up the remaining variance}
  $x \leftarrow \mu + \sqrt{v - v_0}\xi$, where $\xi \sim \mathcal{N}(0, I_d)$
  $r \leftarrow x - Q^d(x)$
  for all i **do**
    sample c_i from $\text{Cat}(|r_i|, v_0)$ as in Equation (5)
  end for
  $\theta \leftarrow Q^d(x) + \text{sign}(r) \odot c$
else {sample from the discrete grid to achieve the target variance}
  $r \leftarrow \mu - Q^s(\mu)$
  for all i **do**
    $v_s \leftarrow \left(1 - \frac{|r_i|}{\Delta}\right) \cdot r_i^2 + \frac{|r_i|}{\Delta} \cdot (-r_i + \text{sign}(r_i)\Delta)^2$
    if $v > v_s$ **then**
      sample c_i from $\text{Cat}(0, v - v_s)$ as in Equation (5)
      $\theta_i \leftarrow Q^s(\mu)_i + c_i$
    else
      $\theta_i \leftarrow Q^s(\mu)_i$
    end if
  end for
end if
clip θ if outside representable range
return θ

This theorem shows that when the stepsize $\alpha \to 0$, VC SGLDLP-L converges to the target distribution up to an error instead of diverging. Moreover, VC SGLDLP-L converges to the target distribution in $\mathcal{O}(\sqrt{\Delta_W})$ which is equivalent to the convergence rate of SGD with low-precision gradient accumulators to the optimum (Li et al., 2017; Yang et al., 2019). However, we show empirically that VC SGLDLP-L has a much better dependency on the quantization gap than SGD. We leave the improvement of the theoretical bound for future work.

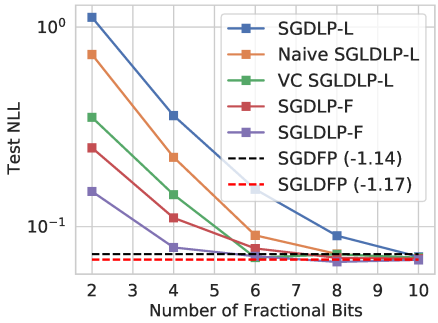
We empirically demonstrate VC SGLDLP-L on the standard Gaussian distribution under the same setting as in the previous section in Figure 1. Regardless of the stepsize, VC SGLDLP-L converges to the target distribution and approximates the target distribution as accurately as SGLDLP-F, showing that preserving the correct variance is the key to ensuring correct convergence.

5. Experiments

We demonstrate the generalization accuracy and uncertainty estimation of low-precision SGLD with full-precision gradi-



(a) Logistic regression



(b) MLP

Figure 2. Test NLL on MNIST in terms of different precision. As the quantization level increases, VC SGLDLP-L and SGLDLP-F are able to maintain high performance whereas the corresponding SGD deteriorates quickly.

ent accumulators (SGLDLP-F) and with variance-corrected low-precision gradient accumulators (VC SGLDLP-L) on a logistic regression and multilayer perceptron on MNIST dataset (Section 5.1), ResNet-18 on CIFAR datasets and LSTM on IMDB dataset (Section 5.2), and ResNet-18 on ImageNet dataset (Section 5.3). We use `qtorch` (Zhang et al., 2019) to simulate low-precision training on these experiments, and use the same quantization for weights, gradients, activations, backpropagation errors unless otherwise stated. For all experiments, SGLD collects samples from the posterior of the model's weight, and obtained the prediction on test data by Bayesian model averaging. We mainly compare low-precision SGLD with low-precision SGD which has been used to achieve state-of-the-art results in low-precision deep learning (Sun et al., 2019; 2020). We use SGLDFP and SGDFP to denote SGLD and SGD in full-precision respectively.

5.1. MNIST

Logistic Regression We first empirically verify our theorems, including the bias dependence on the quantization levels and the tightness of the bounds, on a logistic regression on MNIST dataset. We use $\mathcal{N}(0, 1/6)$ as the prior distribution following Yang et al. (2019) and the resulting posterior distribution is strongly log-concave and satisfies the assumptions in Section 4. We use fixed point numbers with 2 integer bits and vary the number of fractional bits which is corresponding to varying the quantization gap Δ. We report test negative log-likelihood (NLL) with different numbers of fractional bits in Figure 2a.

From the results, we see that SGLDLP-F is more robust to the decay of the number of bits than its SGD counterpart since SGLDLP-F outperforms SGDLP-F on all number of bits and recovers the full-precision result with 6 bits whereas SGDLP-F needs 10 bits. This verifies Theorem 1 that SGLDLP-F converges to the target distribution up to an error and is more robust to the quantization gap than SGDLP-F. This also shows that the bound in Theorem 1 is relatively tight in terms of the quantization gap since SGLDLP-F needs roughly half number of bits compared to SGDLP-F to achieve the same convergence accuracy. With low-precision gradient accumulators, we can see that VC SGLDLP-L is significantly better than naïve SGLDLP-L, indicating that our variance-corrected quantization effectively reduces the bias of gradient accumulators, which verifies Theorem 2 and Theorem 3. Moreover, VC SGLDLP-L outperforms SGDLP-L on all numbers of bits and even outperforms SGDLP-F when using 2 fractional bits. These observations demonstrate that with either full- or low-precision gradient accumulators, SGLD is able to maintain its high performance whereas SGD deteriorates quickly as the quantization noise increases.

Multilayer Perceptron To test whether these results apply to non-log-concave distributions, we replace the logistic regression model by a two-layer multilayer perceptron (MLP). The MLP has 100 hidden units and RELU nonlinearities. From Figure 2b, we observe similar results as on the logistic regression, suggesting that empirically our analysis still stands on general distributions and sheds light on the possibility of extending the theoretical analysis of low-precision SGLD to non-log-concave settings. Please note that as far as we understand, there is no theoretical convergence result of low-precision optimization in non-convex settings either.

5.2. CIFAR and IMDB

We consider image and sentiment classification tasks: CIFAR datasets (Krizhevsky et al., 2009) on ResNet-18 (He et al., 2016), and IMDB dataset (Maas et al., 2011) on LSTM (Hochreiter & Schmidhuber, 1997). We use 8-bit number representation since it becomes increasingly popular in training deep models and is powered by the new generation of chips (Sun et al., 2019; Banner et al., 2018; Wang et al., 2018b). We report test errors averaged over 3 runs with the standard error in Table 1.

Table 1. Test errors (%) on CIFAR with ResNet-18 and IMDB with LSTM. Low-precision SGLD outperforms low-precision SGD across different datasets, architectures and number representations, and the improvement becomes larger when using more low-precision arithmetic.

	CIFAR-10	CIFAR-100	IMDB
32-BIT FLOATING POINT			
SGLDFP	4.65 ±0.06	22.58 ±0.18	13.43 ±0.21
SGDFP	4.71 ±0.02	22.64 ±0.13	13.88 ±0.29
CSGLDFP	4.54 ±0.05	21.63 ±0.04	13.25 ±0.18
8-BIT FIXED POINT			
NAÏVE SGLDLP-L	7.82 ±0.13	27.25 ±0.13	16.63 ±0.28
VC SGLDLP-L	7.13 ±0.01	26.62 ±0.16	15.38 ±0.27
SGDLP-L	8.53 ±0.08	28.86 ± 0.10	19.28 ±0.63
SGLDLP-F	5.12 ±0.06	23.30 ±0.09	15.40 ±0.36
SGDLP-F	5.20 ±0.14	23.84 ±0.12	15.74 ±0.79
8-BIT BLOCK FLOATING POINT			
NAÏVE SGLDLP-L	5.85 ±0.04	26.38 ±0.13	14.64 ±0.08
VC SGLDLP-L	5.51 ±0.01	25.22 ±0.18	13.99 ±0.24
SGDLP-L	5.86 ±0.18	26.19 ±0.11	16.06 ±1.81
SGLDLP-F	4.58 ±0.07	22.59 ±0.18	14.05 ±0.33
SGDLP-F	4.75 ±0.05	22.9 ±0.13	14.28 ±0.17
VC CSGLDLP-L	4.97 ±0.10	22.61 ±0.12	13.09 ±0.27
CSGLD-F	4.32 ±0.07	21.50 ±0.14	13.13 ±0.37

Table 2. ECE ↓ (%) on CIFAR with ResNet-18. VC SGLDLP-L and SGLDLP-F achieve almost the same or even lower ECE than full-precision SGLD whereas the ECE of low-precision SGD increases significantly.

	CIFAR-10	CIFAR-100
32-BIT FLOATING POINT		
SGLD	1.11	3.92
SGD	2.53	4.97
CSGLDFP	0.66	1.38
8-BIT FIXED POINT		
VC SGLDLP-L	0.6	3.19
SGDLP-L	3.4	10.38
SGLDLP-F	1.12	4.42
SGDLP-F	3.05	6.80
8-BIT BLOCK FLOATING POINT		
VC SGLDLP-L	0.6	5.82
SGDLP-L	4.23	12.97
SGLDLP-F	1.19	3.78
SGDLP-F	2.76	5.2
VC CSGLDLP-L	0.51	1.39
CSGLD-F	0.56	1.33

Fixed Point We use 8-bit fixed point for weights and gradients but full-precision for activations since we find low-precision activations significantly harm the performance. Similar to the results in previous sections, SGLDLP-F is better than SGDLP-F and VC SGLDLP-L significantly outperforms naïve SGLDLP-L and SGDLP-L across datasets and architectures. Notably, the improvement of SGLD over SGD becomes larger when using more low-precision arithmetic. For example, on CIFAR-100, VC SGLDLP-L outperforms SGDLP-L by 2.24%, SGLDLP-F outperforms SGDLP-F by 0.54% and SGLDFP outperforms SGDFP by 0.06%. This demonstrates that SGLD is particularly compatible with low-precision deep learning because of its natural ability to handle system noise.

Block Floating Point We also consider block floating point (BFP) which is another common number type and is often preferred over fixed point on deep models due to less quantization error caused by overflow and underflow (Song et al., 2018). Following the block design in Yang et al. (2019), we use *small-block* for ResNet and *big-block* for LSTM. The Q^{vc} function naturally generalizes to BFP and only needs a small modification (see Appendix G for the algorithm of Q^{vc} with BFP). By using BFP, the results of all low-precision methods improve over fixed point. SGLDLP-F can match the performance of SGLDFP with all numbers quantized to 8-bit except gradient accumulators. VC SGLDLP-L still outperforms naïve SGLDLP-L indicating the effectiveness of Q^{vc} with BFP. Again, SGLDFP-F and VC SGLDLP-L outperform their SGD counterparts on all tasks, suggesting the general applicability of low-precision SGLD with different number types.

Cyclical SGLD We further apply low-precision to a recent variant of SGLD, *cSGLD*, which utilizes a cyclical learning rate schedule to speed up convergence (Zhang et al., 2020). We observe that the results of cSGLD-F are very close to those of cSGLDFP, and VC cSGLDLP-L can match or even outperforms full-precision SGD with all numbers quantized to 8 bits! These results indicate that diverse samples from different modes, obtained by the cyclical learning rate schedule, can counter-effect the quantization error by providing complementary predictions.

Expected Calibration Error Besides generalization performance, we further report the results of expected calibration error (ECE) (Guo et al., 2017) to demonstrate the uncertainty estimation of low-precision SGLD. In Table 2, we observe that SGLDLP-F and VC SGLDLP-L achieve almost the same or even lower ECE than full-precision SGLD, showing the ability of SGLD to give well-calibrated predictions does not degenerate due to using low-precision. VC SGLDLP-L sometimes gives lower ECE than SGLDLP-F which may be due to the regularization effect of low-precision arithmetic. Moreover, cSGLD in low-precision not only achieves the best accuracy but also has the best calibration, further suggesting that diverse samples obtained by a cyclical learning rate schedule have a positive effect on quantization. In contrast, the ECE of low-precision SGD increases significantly compared to full-precision SGD, im-

Table 3. Test errors (%) on ImageNet with ResNet-18. The improvement of SGLD over SGD becomes larger in low-precision than in full-precision.

	TOP-1	TOP-5
32-BIT FLOATING POINT		
SGLD	30.39	10.76
SGD	30.56	10.97
8-BIT BLOCK FLOATING POINT		
SGLDLP-F	31.47	11.77
SGDLP-F	32.23	12.09

plying that the quantization error makes the standard DNNs even more overconfident, which might lead to wrong decisions in real-world applications.

SGLDLP-F vs VC SGLDLP-L We have provided two variants of low-precision SGLD for practical use. In general, SGLDLP-F has better performance while VC SGLDLP-L requires less computation, making them suitable for different cases. When the computation resources are very limited, e.g. on edge devices, VC SGLDLP-L is preferred for saving computation while when the resources are able to support full-precision gradient accumulators, SGLDLP-F is preferred for better performance.

5.3. ImageNet

Finally, we test low-precision SGLD on a large-scale image classification dataset, ImageNet, with ResNet-18. We train SGD for 90 epochs and train SGLD for 10 epochs using the trained SGD model as the initialization. In Table 3, we observe that the improvement of SGLD over SGD is larger in low-precision (0.76% top-1 error) than in full-precision (0.17% top-1 error), showing the advantages of low-precision SGLD on large-scale deep learning tasks. We could not achieve reasonable results with low-precision gradient accumulators for SGD and SGLD, which might be caused by hyper-parameter tuning.

6. Conclusion

We provide the first comprehensive investigation for low-precision SGLD. With full-precision gradient accumulators, we prove that SGLD is convergent and can be safely used in practice, and further show that it has a better dependency of convergence on the quantization gap than SGD. Moreover, we reveal issues in naïvely performing low-precision computation in SGLD with low-precision gradient accumulators, and propose a new theoretically guaranteed quantization function to enable fully quantized sampling. We conduct experiments on a Gaussian distribution and a logistic regression to empirically verify our theoretical results. Besides, we show that low-precision SGLD achieves comparable results with full-precision SGLD and outperforms low-precision SGD significantly on several Bayesian deep learning benchmarks.

MCMC was once the gold standard on small neural networks (Neal et al., 2011), but has been significantly limited by its high costs on large architectures in deep learning. We believe this work fills an important gap, and will accelerate the practical use of sampling methods on large-scale and resource-restricted machine learning problems.

Moreover, low-precision SGLD could broadly be used as a drop-in replacement for standard SGLD, as it can confer speed and memory advantages, while retaining accuracy.

Acknowledgements

RZ is supported by NSF AI Institute for Foundations of Machine Learning (IFML). AGW is supported by NSF CAREER IIS-2145492, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, Meta Core Data Science, Google AI Research, BigHat Biosciences, Capital One, and an Amazon Research Award.

References

Achterhold, J., Koehler, J. M., Schmeink, A., and Genewein, T. Variational network quantization. In *International Conference on Learning Representations*, 2018.

Ahn, S., Shahbaba, B., and Welling, M. Distributed stochastic gradient mcmc. In *International conference on machine learning*, pp. 1044–1052. PMLR, 2014.

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. Control variates for stochastic gradient mcmc. *Statistics and Computing*, 29(3):599–615, 2019.

Banner, R., Hubara, I., Hoffer, E., and Soudry, D. Scalable methods for 8-bit training of neural networks. *Advances in neural information processing systems*, 2018.

Cai, R., Ren, A., Liu, N., Ding, C., Wang, L., Qian, X., Pedram, M., and Wang, Y. Vibnn: Hardware acceleration of bayesian neural networks. *ACM SIGPLAN Notices*, 53 (2):476–488, 2018.

Chen, C., Ding, N., Li, C., Zhang, Y., and Carin, L. Stochastic gradient mcmc with stale gradients. *Advances in Neural Information Processing Systems*, 2016.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Cheng, Z., Soudry, D., Mao, Z., and Lan, Z. Training binary multilayer neural networks for image classification using expectation backpropagation. *arXiv preprint arXiv:1503.03562*, 2015.

Courbariaux, M., Bengio, Y., and David, J.-P. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pp. 3123–3131, 2015.

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

De Sa, C., Feldman, M., Ré, C., and Olukotun, K. Understanding and optimizing asynchronous low-precision stochastic gradient descent. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pp. 561–574, 2017.

Deng, W., Lin, G., and Liang, F. A contour stochastic gradient langevin dynamics algorithm for simulations of multi-modal distributions. *Advances in neural information processing systems*, 33:15725–15736, 2020.

Dubey, K. A., J Reddi, S., Williamson, S. A., Poczos, B., Smola, A. J., and Xing, E. P. Variance reduction in stochastic gradient langevin dynamics. *Advances in neural information processing systems*, 29:1154–1162, 2016.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. *International Conference on Learning Representations*, 2020.

Ferianc, M., Maji, P., Mattina, M., and Rodrigues, M. On the effects of quantisation on model uncertainty in bayesian neural networks. *Uncertainty in Artificial Intelligence*, 2021.

Gan, Z., Li, C., Chen, C., Pu, Y., Su, Q., and Carin, L. Scalable bayesian learning of recurrent neural networks for language modeling. *Association for Computational Linguistics*, 2017.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. Deep learning with limited numerical precision. In *International conference on machine learning*, pp. 1737–1746. PMLR, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heek, J. and Kalchbrenner, N. Bayesian inference for large scale image classification. *arXiv preprint arXiv:1908.03491*, 2019.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

Korattikara, A., Rathod, V., Murphy, K., and Welling, M. Bayesian dark knowledge. *arXiv preprint arXiv:1506.04416*, 2015.

Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, C., Stevens, A., Chen, C., Pu, Y., Gan, Z., and Carin, L. Learning weight uncertainty with stochastic gradient mcmc for shape classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5666–5675, 2016.

Li, C., Chen, C., Pu, Y., Henao, R., and Carin, L. Communication-efficient stochastic gradient mcmc for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4173–4180, 2019.

Li, H., De, S., Xu, Z., Studer, C., Samet, H., and Goldstein, T. Training quantized nets: A deeper understanding. *Advances in neural information processing systems*, 2017.

Li, Z. and De Sa, C. M. Dimension-free bounds for low-precision training. *Advances in Neural Information Processing Systems*, 2019.

Lin, D., Talathi, S., and Annapureddy, S. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pp. 2849–2858. PMLR, 2016.

Ma, Y.-A., Chen, T., and Fox, E. B. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 2015.

Ma, Y.-A., Chen, Y., Jin, C., Flammarion, N., and Jordan, M. I. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42): 20881–20885, 2019.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Meng, X., Bachmann, R., and Khan, M. E. Training binary neural networks using the bayesian learning rule. In *International Conference on Machine Learning*, pp. 6852–6861. PMLR, 2020.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. Mixed precision training. *International Conference on Learning Representations*, 2018.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Song, Z., Liu, Z., and Wang, D. Computation error analysis of block floating point arithmetic oriented convolution neural network accelerator design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Soudry, D., Hubara, I., and Meir, R. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *Advances in Neural Information Processing Systems*, volume 1, pp. 2, 2014.

Su, J., Cvitkovic, M., and Huang, F. Sampling-free learning of bayesian quantized neural networks. *arXiv preprint arXiv:1912.02992*, 2019.

Sun, X., Choi, J., Chen, C.-Y., Wang, N., Venkataramani, S., Srinivasan, V. V., Cui, X., Zhang, W., and Gopalakrishnan, K. Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks. *Advances in neural information processing systems*, 32:4900–4909, 2019.

Sun, X., Wang, N., Chen, C.-Y., Ni, J., Agrawal, A., Cui, X., Venkataramani, S., El Maghraoui, K., Srinivasan, V. V., and Gopalakrishnan, K. Ultra-low precision 4-bit training of deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Talwar, K. Computational separations between sampling and optimization. *Advances in neural information processing systems*, 2019.

van Baalen, M., Louizos, C., Nagel, M., Amjad, R. A., Wang, Y., Blankevoort, T., and Welling, M. Bayesian bits: Unifying quantization and pruning. *Advances in Neural Information Processing Systems*, 2020.

Wang, K.-C., Vicol, P., Lucas, J., Gu, L., Grosse, R., and Zemel, R. Adversarial distillation of bayesian neural network posteriors. In *International Conference on Machine Learning*, pp. 5190–5199. PMLR, 2018a.

Wang, N., Choi, J., Brand, D., Chen, C.-Y., and Gopalakrishnan, K. Training deep neural networks with 8-bit floating point numbers. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7686–7695, 2018b.

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

Wu, S., Li, G., Chen, F., and Shi, L. Training and inference with integers in deep neural networks. *International Conference on Learning Representations*, 2018.

Yang, G., Zhang, T., Kirichenko, P., Bai, J., Wilson, A. G., and De Sa, C. Swalp: Stochastic weight averaging in low-precision training. *International Conference on Machine Learning*, 2019.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical stochastic gradient mcmc for bayesian deep learning. *International Conference on Learning Representations*, 2020.

Zhang, T., Lin, Z., Yang, G., and De Sa, C. Qpytorch: A low-precision arithmetic simulation framework. *arXiv preprint arXiv:1910.04540*, 2019.

Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., and Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

Zhu, S., Dong, X., and Su, H. Binary ensemble neural network: More bits per network or more networks per bit? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4923–4932, 2019.

A. Quantization Formulation

We follow the quantization framework in prior work (Wu et al., 2018; Wang et al., 2018b; Yang et al., 2019) to quantize weights, activations, backpropagation errors, and gradients, as outlined in Algorithm 3.

Algorithm 3 Low-Precision Training for SGLD.

given: L layers DNN $\{f_1 \ldots, f_L\}$. Stepsize α. Weight, gradient, activation and error quantizers Q_W, Q_G, Q_A, Q_E. Variance-corrected quantization Q^{vc}, deterministic rounding Q^d, stochastic rounding Q^s and quantization gap of weights Δ_W. Data batch sequence $\{(x_k, y_k)\}_{k=1}^K$. θ_k^{fp} denotes the full-precision buffer of the weight.

for $k = 1 : K$ **do**

1. Forward Propagation:

$a_k^{(0)} = x_k$

$a_k^{(l)} = Q_A(f_l(a_k^{(l-1)}, \theta_k^l)), \forall l \in [1, L]$

2. Backward Propagation:

$e^{(L)} = \nabla_{a_k^{(L)}} \mathcal{L}(a_k^{(L)}, y_k)$

$e^{(l-1)} = Q_E\left(\frac{\partial f_l(a_k^{(l)})}{\partial a_k^{(l-1)}} e_k^{(l)}\right), \forall l \in [1, L]$

$g_k^{(l)} = Q_G\left(\frac{\partial f_l}{\partial \theta_k^{(l)}} e_k^{(l)}\right), \forall l \in [1, L]$

3. SGLD Update:

full-precision gradient accumulators: $\theta_{k+1}^{fp} \leftarrow \theta_k^{fp} - \alpha Q_G\left(\nabla \tilde{U}(\theta_k)\right) + \sqrt{2\alpha}\xi, \quad \theta_{k+1} \leftarrow Q_W\left(\theta_{k+1}^{fp}\right)$

low-precision gradient accumulators: $\theta_{k+1} \leftarrow Q^{vc}\left(\theta_k - \alpha Q_G\left(\nabla \tilde{U}(\theta_k)\right), 2\alpha, \Delta_W\right)$

end for

output: samples $\{\theta_k\}$

B. Proof of Theorem 1

Our proofs in the paper follow Theorem 4 in Dalalyan & Karagulyan (2019), which provides a convergence bound of Langevin dynamics with noisy gradients. We state the result of Theorem 4 in Dalalyan & Karagulyan (2019) below.

We consider Langevin dynamics whose update rule is

$$\theta_{k+1} = \theta_k - \alpha\left(\nabla U(\theta_k) + \zeta_k\right) + \sqrt{2\alpha}\xi_{k+1}. \tag{6}$$

The noise in the gradient ζ_k has the following three assumptions:

$$\mathbf{E}\left[\left\|\mathbf{E}\left[\zeta_k|\theta_k\right]\right\|_2^2\right] \leq \delta^2 d, \quad \mathbf{E}\left[\left\|\zeta_k - \mathbf{E}\left[\zeta_k|\theta_k\right]\right\|_2^2\right] \leq \sigma^2 d, \quad \xi_{k+1} \text{ is independent of } (\zeta_0, \cdots, \zeta_k),$$

where $\delta > 0$ and $\sigma > 0$ are some constants. Under the same assumptions in Section 4, we have the convergence bound for the above Langevin dynamics.

Theorem 4 (Theorem 4 in Dalalyan & Karagulyan (2019))**.** *We run the above Langevin dynamics with $\alpha \leq 2/(m + M)$. Let π be the target distribution, μ_0 be the initial distribution and μ_K be the distribution obtained by the Langevin dynamics in Equation (6) after K iterations. Then*

$$W_2(\mu_K, \pi) \leq (1 - \alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \frac{\delta\sqrt{d}}{m} + \frac{\sigma^2(\alpha d)^{1/2}}{1.65M + \sigma\sqrt{m}}.$$

Buit upon this theorem, we now prove Theorem 1.

Proof. We write the SGLDLP-F update as the following

$$\begin{aligned}\theta_{k+1} &= \theta_k - \alpha Q_G(\nabla \tilde{U}(Q_W(\theta_k))) + \sqrt{2\alpha}\xi_{k+1} \\ &= \theta_k - \alpha\left(\nabla U(\theta_k) + \zeta_k\right) + \sqrt{2\alpha}\xi_{k+1}\end{aligned}$$

where

$$\begin{aligned}\zeta_k &= Q_G(\nabla\tilde{U}(Q_W(\theta_k))) - \nabla U(\theta_k)\\ &= Q_G(\nabla\tilde{U}(Q_W(\theta_k))) - \nabla\tilde{U}(Q_W(\theta_k))\\ &\quad + \nabla\tilde{U}(Q_W(\theta_k)) - \nabla U(Q_W(\theta_k)) + \nabla U(Q_W(\theta_k)) - \nabla U(\theta_k).\end{aligned}$$

Since $\mathbf{E}[\nabla\tilde{U}(x)] = \nabla U(x)$ and $\mathbf{E}[Q(x)] = x$, we have

$$\begin{aligned}\mathbf{E}\left[\zeta_k|\theta_k\right] &= \mathbf{E}\left[Q_G(\nabla\tilde{U}(Q_W(\theta_k))) - \nabla\tilde{U}(Q_W(\theta_k))\Big|\theta_k\right]\\ &\quad + \mathbf{E}\left[\nabla\tilde{U}(Q_W(\theta_k)) - \nabla U(Q_W(\theta_k))\Big|\theta_k\right] + \mathbf{E}\left[\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)|\theta_k\right]\\ &= \mathbf{E}\left[\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)|\theta_k\right].\end{aligned}$$

By the assumption, we know that

$$\left\|\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)\right\|_2^2 \leq M^2\left\|Q_W(\theta_k) - \theta_k\right\|_2^2,$$

then it follows that

$$\begin{aligned}\left\|\mathbf{E}\left[\zeta_k|\theta_k\right]\right\|_2^2 &= \left\|\mathbf{E}\left[\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)|\theta_k\right]\right\|_2^2\\ &\leq \mathbf{E}\left[\left\|\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)\right\|_2^2\Big|\theta_k\right]\\ &\leq M^2\mathbf{E}\left[\left\|Q_W(\theta_k) - \theta_k\right\|_2^2\Big|\theta_k\right]\\ &\leq M^2\cdot\frac{\Delta_W^2 d}{4}.\end{aligned}$$

Let $f:\mathbb{R}\to\mathbb{R}^d$ denote the function

$$f(a) = \nabla U(\theta_k + a(Q_W(\theta_k) - \theta_k)).$$

By the mean value theorem, there will exist an $a\in[0,1]$ (a function of the weight quantization randomness) such that

$$f(1) - f(0) = f'(a).$$

So,

$$\begin{aligned}\mathbf{E}\left[\zeta_k|\theta_k\right] &= \mathbf{E}\left[\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)|\theta_k\right]\\ &= \mathbf{E}\left[\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k))(Q_W(\theta_k) - \theta_k)\big|\theta_k\right]\\ &= \mathbf{E}\left[\nabla^2 U(\theta_k)(Q_W(\theta_k) - \theta_k)\big|\theta_k\right]\\ &\quad + \mathbf{E}\left[\left(\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k)) - \nabla^2 U(\theta_k)\right)(Q_W(\theta_k) - \theta_k)\big|\theta_k\right]\\ &= \mathbf{E}\left[\left(\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k)) - \nabla^2 U(\theta_k)\right)(Q_W(\theta_k) - \theta_k)\big|\theta_k\right].\end{aligned}$$

Now, by the assumption $\|\nabla^2 U(x) - \nabla^2 U(y)\|_2 \leq \Psi\|x-y\|_2$, we get

$$\begin{aligned}\left\|\mathbf{E}\left[\zeta_k|\theta_k\right]\right\|_2 &= \left\|\mathbf{E}\left[\left(\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k)) - \nabla^2 U(\theta_k)\right)(Q_W(\theta_k) - \theta_k)\big|\theta_k\right]\right\|_2\\ &\leq \mathbf{E}\left[\left\|\left(\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k)) - \nabla^2 U(\theta_k)\right)(Q_W(\theta_k) - \theta_k)\right\|_2\big|\theta_k\right]\\ &\leq \mathbf{E}\left[\left\|\nabla^2 U(\theta_k + a(Q_W(\theta_k) - \theta_k)) - \nabla^2 U(\theta_k)\right\|_2\left\|Q_W(\theta_k) - \theta_k\right\|_2\big|\theta_k\right]\\ &\leq \mathbf{E}\left[\Psi\left\|a(Q_W(\theta_k) - \theta_k)\right\|_2\left\|Q_W(\theta_k) - \theta_k\right\|_2|\theta_k\right]\\ &\leq \Psi\mathbf{E}\left[\left\|Q_W(\theta_k) - \theta_k\right\|_2^2\Big|\theta_k\right]\\ &\leq \frac{\Psi\Delta_W^2 d}{4}.\end{aligned}$$

This combined with the previous result gives us

$$\|\mathbf{E}\left[\zeta_k|\theta_k\right]\|_2 \leq \min\left(\frac{\Psi\Delta_W^2 d}{4}, \frac{M\Delta_W\sqrt{d}}{2}\right).$$

Now considering the variance of ζ_k,

$$\begin{aligned}
&\mathbf{E}\left[\|\zeta_k - \mathbf{E}\left[\zeta_k|\theta_k\right]\|_2^2\right] \\
&\quad\leq \mathbf{E}\left[\|\zeta_k\|_2^2\right] \\
&\quad\leq \mathbf{E}\left[\left\|Q_G(\nabla\tilde{U}(Q_W(\theta_k))) - \nabla\tilde{U}(Q_W(\theta_k))\right\|_2^2\right] \\
&\qquad+ \mathbf{E}\left[\left\|\nabla\tilde{U}(Q_W(\theta_k)) - \nabla U(Q_W(\theta_k))\right\|_2^2\right] + \mathbf{E}\left[\|\nabla U(Q_W(\theta_k)) - \nabla U(\theta_k)\|_2^2\right] \\
&\quad\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + M^2\cdot\frac{\Delta_W^2 d}{4}.
\end{aligned}$$

Recall that to apply the result of Theorem 4 in Dalalyan & Karagulyan (2019), we need

$$\mathbf{E}\left[\|\mathbf{E}\left[\zeta_k|\theta_k\right]\|_2^2\right] \leq \delta^2 d, \quad \mathbf{E}\left[\|\zeta_k - \mathbf{E}\left[\zeta_k|\theta_k\right]\|_2^2\right] \leq \sigma^2 d.$$

We set δ and σ to be

$$\delta = \min\left(\frac{\Psi\Delta_W^2\sqrt{d}}{4}, \frac{M\Delta_W}{2}\right), \quad \sigma^2 d = \frac{\Delta_G^2 d}{4} + \kappa^2 + M^2\cdot\frac{\Delta_W^2 d}{4} = \frac{(\Delta_G^2 + M^2\Delta_W^2)d + 4\kappa^2}{4}.$$

Since ζ_k is independent of the Gaussian noise ξ_i, for $i = 0, \ldots, k+1$, we have shown that the assumptions in Theorem 4 in Dalalyan & Karagulyan (2019) are satisfied. Thus we apply the result in Theorem 4 and get

$$\begin{aligned}
W_2(\mu_K, \pi) &\leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \frac{\delta\sqrt{d}}{m} + \frac{\sigma^2(\alpha d)^{1/2}}{1.65M + \sigma\sqrt{m}} \\
&\leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \frac{\delta\sqrt{d}}{m} + \sqrt{\frac{\sigma^2\alpha d}{m}} \\
&= (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) \\
&\qquad+ \sqrt{\frac{(\Delta_G^2 + M^2\Delta_W^2)\alpha d + 4\alpha\kappa^2}{4m}}.
\end{aligned}$$

Note that we ignore the $1.65M$ in the denominator to further simplify the bound. □

C. Proof of Theorem 2

Proof. Recall that the update of SGLDLP-L is

$$\theta_{k+1} = Q_W\left(\theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)) + \sqrt{2\alpha}\xi_{k+1}\right).$$

To utilize the result in Dalalyan & Karagulyan (2019), we introduce an intermediate dynamic $\psi_{k+1} = \theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)) + \sqrt{2\alpha}\xi_{k+1}$. Therefore $\theta_k = Q_W(\psi_k)$ and

$$\begin{aligned}
\psi_{k+1} &= \theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)) + \sqrt{2\alpha}\xi_{k+1} \\
&= Q_W(\psi_k) - \alpha Q_G(\nabla\tilde{U}(Q_W(\psi_k))) + \sqrt{2\alpha}\xi_{k+1} \\
&= \psi_k - \alpha(\nabla U(\psi_k) + \zeta_k) + \sqrt{2\alpha}\xi_{k+1}
\end{aligned}$$

where

$$\begin{aligned}\zeta_k &= \frac{\psi_k - \theta_k}{\alpha} + Q_G(\nabla\tilde{U}(\theta_k)) - \nabla U(\psi_k) \\ &= \frac{\psi_k - \theta_k}{\alpha} + Q_G(\nabla\tilde{U}(\theta_k)) - \nabla\tilde{U}(\theta_k) + \nabla\tilde{U}(\theta_k) - \nabla U(\theta_k) + \nabla U(\theta_k) - \nabla U(\psi_k).\end{aligned}$$

Similar to the previous proof in Section B, we know that

$$\mathbf{E}\left[\zeta_k|\psi_k\right] = \mathbf{E}\left[\nabla U(\theta_k) - \nabla U(\psi_k)|\psi_k\right] = \mathbf{E}\left[\nabla U(Q_W(\psi_k)) - \nabla U(\psi_k)|\psi_k\right],$$

so

$$\left\|\mathbf{E}\left[\zeta_k|\psi_k\right]\right\|_2 \leq \min\left(\frac{\Psi\Delta_W^2 d}{4}, \frac{M\Delta_W\sqrt{d}}{2}\right),$$

and it suffices to set δ the same as in Section B. On the other hand, the variance will be bounded by

$$\begin{aligned}\mathbf{E}\left[\left\|\zeta_k - \mathbf{E}\left[\zeta_k|\psi_k\right]\right\|_2^2\right] &\leq \mathbf{E}\left[\left\|Q_G(\nabla\tilde{U}(\theta_k)) - \nabla\tilde{U}(\theta_k)\right\|_2^2\right] + \mathbf{E}\left[\left\|\nabla\tilde{U}(\theta_k) - \nabla U(\theta_k)\right\|_2^2\right] \\ &\quad + \mathbf{E}\left[\left\|\frac{\psi_k - \theta_k}{\alpha} + \nabla U(\theta_k) - \nabla U(\psi_k)\right\|_2^2\right] \\ &\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + \mathbf{E}\left[\left\|\nabla F(\psi_k) - \nabla F(\theta_k)\right\|_2^2\right],\end{aligned}$$

where $F(\theta) = \frac{1}{2\alpha}\left\|\theta\right\|_2^2 - U(\theta)$. Observe that since U is m-strongly convex and M-smooth, and $\alpha^{-1} \geq M/2$, F must be α^{-1}-smooth, and so

$$\begin{aligned}\mathbf{E}\left[\left\|\zeta_k - \mathbf{E}\left[\zeta_k|\psi_k\right]\right\|_2^2\right] &\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{1}{\alpha^2}\mathbf{E}\left[\left\|\psi_k - \theta_k\right\|_2^2\right] \\ &\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{\Delta_W^2 d}{4\alpha^2}.\end{aligned}$$

This is essentially replacing the M^2 in the previous analysis with α^{-2}. It suffices to set $\sigma^2 d = \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{\Delta_W^2 d}{4\alpha^2}$.

Supposing the distribution of ψ_{K+1} is ν_K, applying Theorem 4 in Dalalyan & Karagulyan (2019) will give us the rate of

$$\begin{aligned}W_2(\nu_K, \pi) \leq (1-\alpha m)^K W_2(\nu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) \\ + \sqrt{\frac{(\alpha\Delta_G^2 + \alpha^{-1}\Delta_W^2)d + 4\alpha\kappa^2}{4m}}.\end{aligned}$$

We also have

$$W_2(\mu_k, \nu_k) = \left(\inf_{J\in\mathcal{J}(x,y)}\int\left\|x-y\right\|_2^2 dJ(x,y)\right)^{1/2} \leq \mathbf{E}\left[\left\|\theta_{k+1} - \psi_{k+1}\right\|_2^2\right]^{\frac{1}{2}} \leq \frac{\Delta_W\sqrt{d}}{2}.$$

Combining these two results, we get the final bound

$$\begin{aligned}W_2(\mu_K, \pi) &\leq W_2(\mu_K, \nu_K) + W_2(\nu_K, \pi) \\ &\leq (1-\alpha m)^K W_2(\nu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) \\ &\quad + \sqrt{\frac{(\alpha\Delta_G^2 + \alpha^{-1}\Delta_W^2)d + 4\alpha\kappa^2}{4m}} + \frac{\Delta_W\sqrt{d}}{2} \\ &\leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi\Delta_W^2 d}{4m}, \frac{M\Delta_W\sqrt{d}}{2m}\right) \\ &\quad + \sqrt{\frac{(\alpha\Delta_G^2 + \alpha^{-1}\Delta_W^2)d + 4\alpha\kappa^2}{4m}} + \left((1-\alpha m)^K + 1\right)\frac{\Delta_W\sqrt{d}}{2}.\end{aligned}$$

□

D. Proof of Theorem 3

Proof. Recall that the update of VC SGLDLP-L is

$$\theta_{k+1} = Q^{vc}\left(\theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)), 2\alpha, \Delta_W\right).$$

We ignore the variance of Q_G since it is relatively small compared to the weight quantization variance in practice. Q^{vc} is defined as in Algorithm 2 and we have $\mathbf{E}\left[\theta_{k+1}|\theta_k\right] = \theta_k - \alpha\nabla U(\theta_k)$.

Let $\psi_{k+1} = \theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)) + \sqrt{2\alpha}\xi_{k+1}$ then it follows that

$$\psi_{k+1} - \theta_{k+1} = \theta_k - \alpha Q_G(\nabla\tilde{U}(\theta_k)) + \sqrt{2\alpha}\xi_{k+1} - \theta_{k+1},$$

and

$$\psi_{k+1} = \psi_k - \alpha(\nabla U(\psi_k) + \zeta_k) + \sqrt{2\alpha}\xi_{k+1}$$

where

$$\begin{aligned}\zeta_k &= \frac{\psi_k - \theta_k}{\alpha} + Q_G(\nabla\tilde{U}(\theta_k)) - \nabla U(\psi_k)\\ &= \frac{\psi_k - \theta_k}{\alpha} + Q_G(\nabla\tilde{U}(\theta_k)) - \nabla\tilde{U}(\theta_k) + \nabla\tilde{U}(\theta_k) - \nabla U(\theta_k) + \nabla U(\theta_k) - \nabla U(\psi_k).\end{aligned}$$

Note that $\mathbf{E}[\psi_k - \theta_k] = 0$. Similar to the previous proof in Section B, we know that

$$\|\mathbf{E}\left[\zeta_k|\psi_k\right]\|_2^2 = \|\mathbf{E}\left[\nabla U(\theta_k) - \nabla U(\psi_k)|\psi_k\right]\|_2^2 \le M^2\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2\middle|\psi_k\right].$$

When $2\alpha > v_0 = \frac{\Delta_W^2}{4}$, we have that

$$\begin{aligned}&\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2\middle|\psi_k\right]\\ &= \mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1}))\right) + \sqrt{2\alpha}\xi_k - Q^d\left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1})) + \sqrt{2\alpha - v_0}\xi_k\right) - \text{sign}(r)c\right\|_2^2\middle|\psi_k\right].\end{aligned}$$

Let

$$\begin{aligned}b = Q^d&\left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1})) + \sqrt{2\alpha - v_0}\xi_k\right)\\ &- \left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1})) + \sqrt{2\alpha - v_0}\xi_k\right),\end{aligned}$$

then $|b| \le \frac{\Delta_W}{2}$ and

$$\begin{aligned}&\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2\middle|\psi_k\right]\\ &= \mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1}))\right) + \sqrt{2\alpha}\xi_k - \left(\theta_{k-1} - \alpha Q_G(\nabla\tilde{U}(\theta_{k-1})) + \sqrt{2\alpha - v_0}\xi_k\right) - b - \text{sign}(r)c\right\|_2^2\middle|\psi_k\right]\\ &= \mathbf{E}\left[\left\|\sqrt{2\alpha}\xi_k - \sqrt{2\alpha - v_0}\xi_k - b - \text{sign}(r)c\right\|_2^2\middle|\psi_k\right]\\ &\le \mathbf{E}\left[\left\|\sqrt{2\alpha}\xi_k - \sqrt{2\alpha - v_0}\xi_k - b\right\|_2^2\middle|\psi_k\right] + \mathbf{E}\left[\|\text{sign}(r)c\|_2^2\middle|\psi_k\right]\\ &\le \mathbf{E}\left[\left\|\left|\sqrt{2\alpha}\xi_k - \sqrt{2\alpha - v_0}\xi_k\right| + \frac{\Delta_W}{2}\right\|_2^2\right] + v_0 d\\ &\le (\sqrt{2\alpha} - \sqrt{2\alpha - v_0})^2\mathbf{E}[\|\xi_k\|_2^2] + (\sqrt{2\alpha} - \sqrt{2\alpha - v_0})\Delta_W\mathbf{E}[\|\xi_k\|_2] + 2v_0 d\\ &\le \left((\sqrt{2\alpha} - \sqrt{2\alpha - v_0})^2 + (\sqrt{2\alpha} - \sqrt{2\alpha - v_0})\Delta_W + 2v_0\right)d.\end{aligned}$$

Since $2xy \leq x^2 + y^2$, we get

$$(\sqrt{2\alpha} - \sqrt{2\alpha - v_0})\Delta_W \leq (\sqrt{2\alpha} - \sqrt{2\alpha - v_0})^2 + \frac{\Delta_W^2}{4} = (\sqrt{2\alpha} - \sqrt{2\alpha - v_0})^2 + v_0.$$

The expression can be further simplified to be

$$\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2 \middle| \psi_k\right] \leq \left(2(\sqrt{2\alpha} - \sqrt{2\alpha - v_0})^2 + 3v_0\right) d.$$

We also note that

$$\sqrt{2\alpha} - \sqrt{2\alpha - v_0} = \frac{2\alpha - (2\alpha - v_0)}{\sqrt{2\alpha} + \sqrt{2\alpha - v_0}} = \frac{v_0}{\sqrt{2\alpha} + \sqrt{2\alpha - v_0}} \leq \frac{v_0}{\sqrt{2\alpha}},$$

then the expectation becomes

$$\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2 \middle| \psi_k\right] \leq \left(\frac{v_0^2}{\alpha} + 3v_0\right) d.$$

Since $2\alpha > v_0$, it follows that

$$\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2 \middle| \theta_k\right] \leq (2v_0 + 3v_0)\, d = 5v_0 d.$$

Let $A = 5v_0 d$. Then we obtain

$$\|\mathbf{E}\left[\zeta_k | \psi_k\right]\|_2^2 \leq M^2 \cdot A,$$

and

$$\|\mathbf{E}\left[\zeta_k | \psi_k\right]\|_2 \leq \Psi \cdot A.$$

Therefore, it suffices to set

$$\delta = \min\left(\Psi A, M\sqrt{A}\right).$$

We now consider the variance which will be bounded by

$$\begin{aligned}
\mathbf{E}\left[\|\zeta_k - \mathbf{E}\left[\zeta_k | \psi_k\right]\|_2^2\right] &\leq \mathbf{E}\left[\left\|Q_G(\nabla\tilde{U}(\theta_k)) - \nabla\tilde{U}(\theta_k)\right\|_2^2\right] + \mathbf{E}\left[\left\|\nabla\tilde{U}(\theta_k) - \nabla U(\theta_k)\right\|_2^2\right] \\
&\quad + \mathbf{E}\left[\left\|\frac{\psi_k - \theta_k}{\alpha} + \nabla U(\theta_k) - \nabla U(\psi_k)\right\|_2^2\right] \\
&\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{1}{\alpha^2}\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2\right] \\
&\leq \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{A}{\alpha^2}.
\end{aligned}$$

It suffices to set $\sigma^2 d = \frac{\Delta_G^2 d}{4} + \kappa^2 + \frac{A}{\alpha^2}$. Supposing the distribution of ψ_{K+1} is ν_K, applying Theorem 4 in Dalalyan & Karagulyan (2019) will give us the rate of

$$\begin{aligned}
W_2(\nu_K, \pi) \leq (1 - \alpha m)^K W_2(\nu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi \cdot A}{m}, \frac{M\sqrt{A}}{m}\right) \\
+ \sqrt{\frac{\alpha\Delta_G^2 d + 4\alpha\kappa^2}{4m} + \frac{A}{\alpha m}}.
\end{aligned}$$

We also have

$$W_2(\mu_K, \nu_K) = \left(\inf_{J \in \mathcal{J}(x,y)} \int \|x - y\|_2^2 \, dJ(x, y)\right)^{1/2} \leq \mathbf{E}\left[\|\theta_{K+1} - \psi_{K+1}\|_2^2\right]^{\frac{1}{2}} \leq \sqrt{A}.$$

Combining these two results, we get

$$\begin{aligned}
W_2(\mu_K, \pi) &\leq W_2(\mu_K, \nu_K) + W_2(\nu_K, \pi) \\
&\leq (1-\alpha m)^K W_2(\nu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi \cdot A}{m}, \frac{M\sqrt{A}}{m}\right) \\
&\quad + \sqrt{\frac{\alpha \Delta_G^2 d + 4\alpha\kappa^2}{4m} + \frac{A}{\alpha m}} + \sqrt{A} \\
&\leq (1-\alpha m)^K W_2(\mu_0, \pi) + 1.65(M/m)(\alpha d)^{1/2} + \min\left(\frac{\Psi \cdot A}{m}, \frac{M\sqrt{A}}{m}\right) \\
&\quad + \sqrt{\frac{\alpha \Delta_G^2 d + 4\alpha\kappa^2}{4m} + \frac{A}{\alpha m}} + \left((1-\alpha m)^K + 1\right)\sqrt{A}.
\end{aligned}$$

When $2\alpha < \frac{\Delta_W^2}{4}$, since we assume that the gradient is bounded by $\mathbf{E}\left[\left\|Q_G(\nabla \tilde{U}(\theta_k))\right\|_1\right] \leq G$,

$$\begin{aligned}
\mathbf{E}[\|\psi_k - \theta_k\|_2^2] &= \mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right) - \theta_k + \sqrt{2\alpha}\xi_k\right\|_2^2\right] \\
&= \mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right) - \theta_k\right\|_2^2\right] + \mathbf{E}\left[\left\|\sqrt{2\alpha}\xi_k\right\|_2^2\right] \\
&\leq \max\left(2\mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right) - Q^s\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right)\right\|_2^2\right], 4\alpha d\right).
\end{aligned}$$

Using the bound equation (6) in Li & De Sa (2019) gives us,

$$\begin{aligned}
&\mathbf{E}\left[\left\|\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right) - Q^s\left(\theta_{k-1} - \alpha Q_G(\nabla \tilde{U}(\theta_{k-1}))\right)\right\|_2^2\right] \\
&\quad \leq \Delta_W \alpha \mathbf{E}\left[\left\|Q_G(\nabla \tilde{U}(\theta_{k-1}))\right\|_1\right] \\
&\quad \leq \Delta_W \alpha G.
\end{aligned}$$

It follows that

$$\mathbf{E}\left[\|\psi_k - \theta_k\|_2^2\right] \leq \max\left(2\Delta_W \alpha G, 4\alpha d\right).$$

Let $A = \max\left(2\Delta_W \alpha G, 4\alpha d\right)$. The rest is that same as in the case $2\alpha > v_0$.

□

E. Comparison to SGD Bounds

There have been many works on comparing optimization and sampling algorithms since they serve as two main computational strategies for machine learning (Ma et al., 2019; Talwar, 2019). For example, in Ma et al. (2019), the authors compare the total variation distance between the approximate distribution and the target distribution (sampling bound), with the objective gap (optimization bound). Following previous work, we compare our 2-Wasserstein distance bound with previous SGD bounds. Previous low-precision SGD convergence bounds are shown in terms of the squared distance to the optimum $\left\|\bar{\theta}_K - \theta^*\right\|_2^2$ (Yang et al., 2019). In order to compare our bounds with theirs, we consider a 2-Wasserstein distance between two point distributions. Let μ_K be the point distribution assigns zero probability everywhere except $\bar{\theta}_K$ and π be the point distribution assigns zero probability everywhere except θ^*. Then we get

$$W_2(\mu_K, \pi) = \left(\inf_{J \in \mathcal{J}(x,y)} \int \|x-y\|_2^2 \, dJ(x,y)\right)^{1/2} \leq \mathbf{E}\left[\left\|\bar{\theta}_K - \theta^*\right\|_2^2\right]^{\frac{1}{2}}.$$

From Yang et al. (2019), we know that $\mathbf{E}\left[\left\|\bar{\theta}_K - \theta^*\right\|_2^2\right]^{\frac{1}{2}}$ is proportional to Δ_W. Therefore, our 2-Wasserstein distance is $O(\Delta_W^2)$ whereas SGD's 2-Wasserstein distance is $O(\Delta_W)$, which shows SGLD is more robust to the quantization error.

F. Deterministic Rounding vs Stochastic Rounding

Compared to deterministic rounding, stochastic rounding is unbiased and thus can preserve gradient information even when the gradient update is smaller than the quantization gap. In theory, deterministic rounding will make the convergence bound worse due to its bias. For example, in Theorem 1, using deterministic rounding as the weight quantizer makes the bias term becomes $\mathcal{O}(\Delta_w)$, which is worse than the current $\mathcal{O}(\Delta_w^2)$. In practice, stochastic rounding generally provides much better results than deterministic rounding especially on deep neural networks (Gupta et al., 2015; Wang et al., 2018b). For example, on CIFAR10 with 8-bit block floating point, we found that using deterministic rounding to quantize the weight in SGDLP-L and SGLDLP-L gives test errors 7.44% and 7.37% respectively, which are much worse than using stochastic rounding (SGDLP-L:5.86%, naïve SGLDLP-L: 5.85%, VC SGLDLP-L: 5.51%).

G. Algorithms with (Block) Floating Point Numbers

The qunatization gaps in floating point and block floating point change depending on the number values. Therefore, we need to compute the qunatization gap in each step in order to apply our variance-vorrected quantization function Q^{vc}. It is easy to see that the qunatization gap can be computed as

$$\Delta_W(\mu) \leftarrow \begin{cases} 2^{E[\mu]-W+2} \text{ where } E[\mu] = \text{clip}(\lfloor \log_2(\max|\mu|) \rfloor, l, u) & \text{block floating point} \\ 2^{E[\mu]-W} \text{ where } E[\mu] = \text{clip}(\lfloor \log_2(|\mu|) \rfloor, l, u) & \text{floating point.} \end{cases} \tag{7}$$

Deterministic rounding and stochastic rounding are defined using the above Δ_W. Then we obtain Q^{vc} function with (block) floating point in Algorithm 5. This algorithm is the same as Algorithm 1 except the lines in red where we recompute the quantization gap Δ after adding Gaussian noise to make sure it aligns with the quantization gap of x. VC SGLDLP-L with (block) floating point is outlined in Algorithm 4.

Algorithm 4 VC SGLDLP-L with (Block) Floating Point.

given: Stepsize α, number of training iterations K, gradient quantizer Q_G, deterministic rounding with (block) floating point Q^d, stochastic rounding with (block) floating point Q^s, F bits to represent the shared exponent (block floating point) or the exponent (floating point), W bits to represent each number in the block (block floating point) or the mantissa (floating point).
let $l \leftarrow -2^{F-1}, u \leftarrow 2^{F-1} - 1$
for $k = 1 : K$ **do**
  compute $\mu \leftarrow \theta_k - \alpha Q_G\left(\nabla \tilde{U}(\theta_{k-1})\right)$
  compute $\Delta_W(\mu)$ following Equation (7)
  update $\theta_{k+1} \leftarrow Q^{vc}(\mu, 2\alpha, \Delta_W(\mu))$
end for
output: samples $\{\theta_k\}$

Algorithm 5 Variance-Corrected Quantization Function Q^{vc} with (Block) Floating Point.

input: (μ, v, Δ)
$v_0 \leftarrow \Delta^2/4$
if $v > v_0$ **then**
  $x \leftarrow \mu + \sqrt{v - v_0}\xi$, where $\xi \sim \mathcal{N}(0, I_d)$
  $r \leftarrow x - Q^d(x)$
  recompute $\Delta \leftarrow \Delta_W(x)$ following Equation (7)
  $v_0 \leftarrow \Delta^2/4$
  for all i **do**
    sample c_i from Cat$(|r_i|, v_0)$ as in Equation (5)
  end for
  $\theta \leftarrow Q^d(x) + \text{sign}(r) \odot c$
else
  the same as in fixed point numbers
end if

H. Experimental Details and Additional Results

H.1. Sampling methods

For both SGLD and low-precision SGLD, we collected samples $\{\theta_k\}_{K=1}^J$ from the posterior of the model's weight, and obtained the prediction on test data $\{x^*, y^*\}$ by Bayesian model averaging

$$p(y^*|x^*, \mathcal{D}) \approx \frac{1}{J}\sum_{j=1}^{J} p(y^*|x^*, \mathcal{D}, \theta_j).$$

H.2. MNIST

We train all methods on logistic regression and MLP for 20 epochs with learning rate 0.1 and batch size 64. We additionally report test error comparisons in Figure 3.

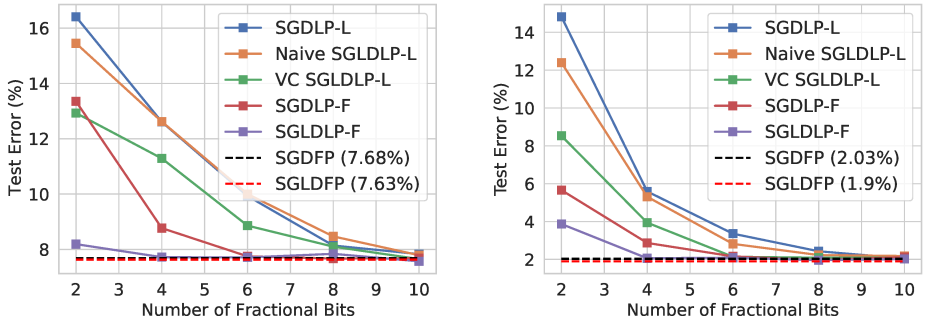


Figure 3. Test error on MNIST dataset in terms of different precision.

H.3. CIFAR and IMDB

For CIFAR datasets, we use batch size 128, learning rate 0.5 and weight decay $5e-4$. We train the model for 245 epochs and used the same decay learning rate schedule as in Yang et al. (2019). We collect 50 samples for SGLD. For cyclical learning rate schedule, we use 7 cycles and collect 5 models per cycle (35 models in total). We use 10 bins for expected calibration error (ECE) following prior work (Guo et al., 2017).

For IMDB dataset, we use batch size 80, learning rate 0.3 and weight decay $5e-4$. We use a two-layer LSTM. The embedding dimension and the hidden dimension are 100 and 256 respectively. We train the model for 50 epochs and used

the same decay learning rate schedule as on CIFAR datasets. We collect 20 samples for SGLD. For cyclical learning rate schedule, we use 1 cycles and collect 20 models.

H.4. ImageNet

We use batch size 256, learning rate 0.2 and weight decay $1e-4$. We use the same decay learning rate schedule as in He et al. (2016) and collect 20 models for SGLD.