Non-convex Learning via Replica Exchange Stochastic Gradient MCMC

A scalable parallel tempering algorithm for DNNs

Wei Deng [1] Qi Feng[*2] Liyao Gao[* 1] Faming Liang [1] Guang Lin [1]

July 27, 2020

[1]Purdue University

[2]University of Southern California

[*]Equal contribution

Intro

The increasing concern for AI safety problems draws our attention to Markov chain Monte Carlo (MCMC), which is known for

- Multi-modal sampling [Teh et al., 2016]
- Non-convex optimization [Zhang et al., 2017]

Acceleration strategies for MCMC

Popular strategies to `accelerate` MCMC:

- Simulated annealing [Kirkpatrick et al., 1983]
- Simulated tempering [Marinari and Parisi, 1992]
- Replica exchange MCMC [Swendsen and Wang, 1986]

Replica exchange stochastic gradient MCMC

Replica exchange Langevin diffusion

Consider two Langevin diffusion processes with $\tau_1 > \tau_2$

$$d\boldsymbol{\beta}_t^{(1)} = -\nabla U(\boldsymbol{\beta}_t^{(1)})dt + \sqrt{2\tau_1}d\boldsymbol{W}_t^{(1)}$$
$$d\boldsymbol{\beta}_t^{(2)} = -\nabla U(\boldsymbol{\beta}_t^{(2)})dt + \sqrt{2\tau_2}d\boldsymbol{W}_t^{(2)},$$

Moreover, the positions of the two particles swap with a probability

$$S(\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)}) := e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(U(\boldsymbol{\beta}_t^{(1)}) - U(\boldsymbol{\beta}_t^{(2)})\right)}$$

In other words, a jump process is included in a Markov process

$$\mathbb{P}(\boldsymbol{\beta}_{t+dt} = (\boldsymbol{\beta}_t^{(2)}, \boldsymbol{\beta}_t^{(1)}) | \boldsymbol{\beta}_t = (\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)})) = rS(\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)})dt$$
$$\mathbb{P}(\boldsymbol{\beta}_{t+dt} = (\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)}) | \boldsymbol{\beta}_t = (\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)})) = 1 - rS(\boldsymbol{\beta}_t^{(1)}, \boldsymbol{\beta}_t^{(2)})dt$$

A demo

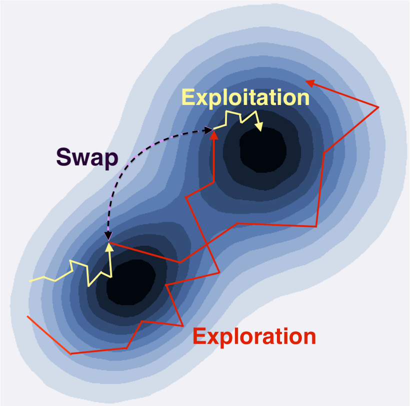


Figure 1: Trajectory plot for replica exchange Langevin diffusion.

Why the naïve numerical algorithm fails

Consider the scalable stochastic gradient Langevin dynamics algorithm [Welling and Teh, 2011]

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_{k}^{(1)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(1)}) + \sqrt{2\eta_k\tau_1}\boldsymbol{\xi}_k^{(1)}$$
$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_{k}^{(2)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(2)}) + \sqrt{2\eta_k\tau_2}\boldsymbol{\xi}_k^{(2)}.$$

Swap the chains with a **naïve** swapping rate $r\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\eta_k$[§]:

$$\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) = e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\right)}. \tag{1}$$

Exponentiating the unbiased estimators $\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(\cdot)})$ leads to a **large bias**.

[§] In the implementations, we fix $r\eta_k = 1$ by default.

Why the naïve numerical algorithm fails

Consider the scalable stochastic gradient Langevin dynamics algorithm [Welling and Teh, 2011]

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_{k}^{(1)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(1)}) + \sqrt{2\eta_k \tau_1}\boldsymbol{\xi}_k^{(1)}$$
$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_{k}^{(2)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(2)}) + \sqrt{2\eta_k \tau_2}\boldsymbol{\xi}_k^{(2)}.$$

Swap the chains with a naïve swapping rate $r\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\eta_k$[§]:

$$\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) = e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\right)}. \tag{1}$$

Exponentiating the unbiased estimators $\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(\cdot)})$ leads to a **large bias**.

[§] In the implementations, we fix $r\eta_k = 1$ by default.

Why the naïve numerical algorithm fails

Consider the scalable stochastic gradient Langevin dynamics algorithm [Welling and Teh, 2011]

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_{k}^{(1)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(1)}) + \sqrt{2\eta_k\tau_1}\boldsymbol{\xi}_k^{(1)}$$
$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_{k}^{(2)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(2)}) + \sqrt{2\eta_k\tau_2}\boldsymbol{\xi}_k^{(2)}.$$

Swap the chains with a **naïve** swapping rate $r\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\eta_k$[§]:

$$\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) = e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\right)}. \tag{1}$$

Exponentiating the unbiased estimators $\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(\cdot)})$ leads to a large bias.

§ In the implementations, we fix $r\eta_k = 1$ by default.

A corrected algorithm

Assume $\widetilde{L}(\boldsymbol{\theta}) \sim \mathcal{N}(L(\boldsymbol{\theta}), \sigma^2)$ and consider the **geometric Brownian motion** of $\{\widetilde{S}_t\}_{t\in[0,1]}$ in each swap as a Martingale

$$\begin{aligned} \widetilde{S}_t &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(1)})-\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t\right)} \\ &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(L(\widetilde{\boldsymbol{\beta}}^{(1)})-L(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t+\sqrt{2}\sigma W_t\right)}. \end{aligned} \tag{2}$$

Taking the derivative of $\widetilde{S}_t$ with respect to t and W_t, Itô's lemma gives,

$$d\widetilde{S}_t = \left(\frac{d\widetilde{S}_t}{dt} + \frac{1}{2}\frac{d^2\widetilde{S}_t}{dW_t^2}\right)dt + \frac{d\widetilde{S}_t}{dW_t}dW_t = \sqrt{2}\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma\widetilde{S}_t dW_t.$$

By fixing $t=1$ in (2), we have the **suggested unbiased swapping rate**

$$\widetilde{S}_1 = e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(1)})-\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\boldsymbol{\sigma}^2\right)}.$$

A corrected algorithm

Assume $\widetilde{L}(\boldsymbol{\theta}) \sim \mathcal{N}(L(\boldsymbol{\theta}), \sigma^2)$ and consider the **geometric Brownian motion** of $\{\widetilde{S}_t\}_{t\in[0,1]}$ in each swap as a Martingale

$$\begin{aligned}\widetilde{S}_t &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(1)})-\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t\right)} \\ &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(L(\widetilde{\boldsymbol{\beta}}^{(1)})-L(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t+\sqrt{2}\sigma W_t\right)}.\end{aligned} \tag{2}$$

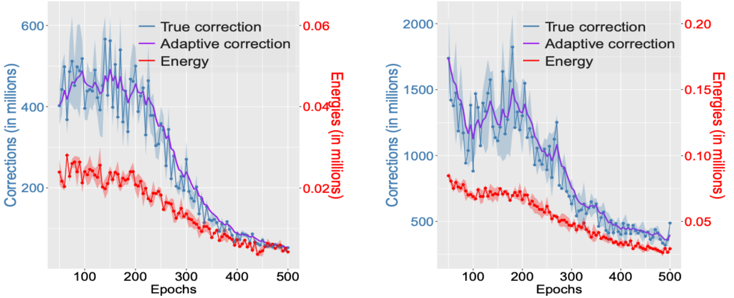
Taking the derivative of $\widetilde{S}_t$ with respect to t and W_t, Itô's lemma gives,

$$d\widetilde{S}_t = \left(\frac{d\widetilde{S}_t}{dt} + \frac{1}{2}\frac{d^2\widetilde{S}_t}{dW_t^2}\right)dt + \frac{d\widetilde{S}_t}{dW_t}dW_t = \sqrt{2}\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma\widetilde{S}_t dW_t.$$

By fixing $t = 1$ in (2), we have the suggested unbiased swapping rate

$$\widetilde{S}_1 = e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(1)})-\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\boldsymbol{\sigma}^2\right)}.$$

Unknown corrections in practice



(a) Time-varying corrections and losses on CIFAR10

(b) Time-varying corrections and losses on CIFAR100

Figure 2: Unknown corrections on CIFAR 10 and CIFAR 100 datasets.

An adaptive algorithm for unknown corrections

Sampling step

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_{k}^{(1)} - \eta_k^{(1)} \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(1)}) + \sqrt{2\eta_k^{(1)}\tau_1}\boldsymbol{\xi}_k^{(1)}$$
$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_{k}^{(2)} - \eta_k^{(2)} \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(2)}) + \sqrt{2\eta_k^{(2)}\tau_2}\boldsymbol{\xi}_k^{(2)},$$

Stochastic approximation step

Obtain an unbiased estimate $\tilde{\sigma}^2_{m+1}$ for σ^2.

$$\hat{\sigma}^2_{m+1} = (1-\gamma_m)\hat{\sigma}^2_m + \gamma_m \tilde{\sigma}^2_{m+1},$$

Swapping step

Generate a uniform random number $u \in [0, 1]$.

$$\hat{S}_1 = \exp\left(\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) - \frac{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\hat{\sigma}^2_{m+1}}{F}\right)\right)$$

If $u < \hat{S}_1$: Swap $\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}$ and $\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}$.

Convergence Analysis

Replica exchange SGLD **tracks** the replica exchange Langevin diffusion in some sense.

Lemma (Discretization Error)

Given the smoothness and dissipativity assumptions in the appendix, and a small (fixed) learning rate η, we have that

$$\mathbb{E}[\sup_{0\leq t\leq T}||\boldsymbol{\beta}_t-\widetilde{\boldsymbol{\beta}}_t^{\eta}||^2]\leq\tilde{\mathcal{O}}(\eta+\max_i\mathbb{E}[\|\boldsymbol{\phi}_i\|^2]+\max_i\sqrt{\mathbb{E}[|\psi_i|^2]}),$$

where $\widetilde{\boldsymbol{\beta}}_t^{\eta}$ is the continuous-time interpolation for reSGLD, $\boldsymbol{\phi} := \nabla\widetilde{U} - \nabla U$ is the noise in the stochastic gradient, and $\psi := \widetilde{S} - S$ is the noise in the stochastic swapping rate.

Accelerated exponential decay of $\mathcal{W}_2$

(i) **Log-Sobolev inequality** for Langevin diffusion [Cattiaux et al., 2010]

Hessian Lower bound
Smooth gradient condition $\rightarrow \nabla^2 G \succcurlyeq -CI_{2d}$ for some constant $C > 0$.

Poincaré inequality
[Chen et al., 2019] $\rightarrow \chi^2(\nu||\pi) \leq c_p \mathcal{E}(\sqrt{\frac{d\nu_t}{d\pi}})$

Lyapunov condition
$V(x_1, x_2) := e^{a/4 \cdot \left(\frac{\|x_1\|^2}{\tau_1} + \frac{\|x_2\|^2}{\tau_2}\right)} \rightarrow \frac{\mathcal{L}V(x_1,x_2)}{V(x_1,x_2)} \leq \kappa - \gamma(\|x_1\|^2 + \|x_2\|^2)$

(ii) Comparison method: acceleration with **a larger Dirichlet form**

$$\mathcal{E}_S(f) = \mathcal{E}(f) + \underbrace{\frac{1}{2}\int S(x_1, x_2) \cdot (f(x_2, x_1) - f(x_1, x_2))^2 d\pi(x_1, x_2),}_{\text{acceleration}}, \quad (3)$$

(i) Log-Sobolev inequality for Langevin diffusion [Cattiaux et al., 2010]

Hessian Lower bound
Smooth gradient condition $\rightarrow \nabla^2 G \succcurlyeq -CI_{2d}$ for some constant $C > 0$.

Poincaré inequality
[Chen et al., 2019] $\rightarrow \chi^2(\nu||\pi) \le c_p \mathcal{E}(\sqrt{\frac{d\nu_t}{d\pi}})$

Lyapunov condition
$V(x_1, x_2) := e^{a/4 \cdot \left(\frac{\|x_1\|^2}{\tau_1} + \frac{\|x_2\|^2}{\tau_2}\right)} \rightarrow \frac{\mathcal{L}V(x_1,x_2)}{V(x_1,x_2)} \le \kappa - \gamma(\|x_1\|^2 + \|x_2\|^2)$

(ii) Comparison method: acceleration with **a larger Dirichlet form**

$$\mathcal{E}_S(f) = \mathcal{E}(f) + \underbrace{\frac{1}{2}\int S(x_1, x_2) \cdot (f(x_2, x_1) - f(x_1, x_2))^2 d\pi(x_1, x_2)}_{\text{acceleration}},, \quad (3)$$

Accelerated exponential decay of $\mathcal{W}_2$

(i) **Log-Sobolev inequality** for Langevin diffusion [Cattiaux et al., 2010]

Hessian Lower bound
Smooth gradient condition $\rightarrow \nabla^2 G \succcurlyeq -CI_{2d}$ for some constant $C > 0$.

Poincaré inequality
[Chen et al., 2019] $\rightarrow \chi^2(\nu||\pi) \leq c_p \mathcal{E}(\sqrt{\frac{d\nu_t}{d\pi}})$

Lyapunov condition
$V(x_1, x_2) := e^{a/4\cdot\left(\frac{\|x_1\|^2}{\tau_1}+\frac{\|x_2\|^2}{\tau_2}\right)} \rightarrow \frac{\mathcal{L}V(x_1,x_2)}{V(x_1,x_2)} \leq \kappa - \gamma(\|x_1\|^2 + \|x_2\|^2)$

(ii) Comparison method: acceleration with a larger Dirichlet form

$$\mathcal{E}_S(f) = \mathcal{E}(f) + \underbrace{\frac{1}{2}\int S(x_1, x_2) \cdot (f(x_2, x_1) - f(x_1, x_2))^2 d\pi(x_1, x_2)}_{\text{acceleration}}, , \quad (3)$$

Theorem (Convergence of reSGLD)
Let the smoothness and dissipativity assumptions hold. For the distribution $\{\mu_k\}_{k\geq 0}$ *associated with the discrete dynamics* $\{\widetilde{\boldsymbol{\beta}}_k\}_{k\geq 1}$*, we have the following estimates, for* $k \in \mathbb{N}^+$*,*

$$\mathcal{W}_2(\mu_k,\pi) \leq D_0 e^{-k\eta(1+\delta_S)/c_{LS}} + \tilde{\mathcal{O}}(\eta^{\frac{1}{2}} + \max_i(\mathbb{E}[\|\phi_i\|^2])^{\frac{1}{2}} + \max_i(\mathbb{E}\left[|\psi_i|^2\right])^{\frac{1}{4}}),$$

where $\delta_S = \min_i \frac{\mathcal{E}_S(\sqrt{\frac{d\mu_i}{d\pi}})}{\mathcal{E}(\sqrt{\frac{d\mu_i}{d\pi}})} - 1$ *is the* ***acceleration effect*** *depending on the swapping rate S,* $D_0 = \sqrt{2c_{LS}D(\mu_0||\pi)}$*,* $\delta_S := \min_i \frac{\mathcal{E}_S(\sqrt{\frac{d\mu_i}{d\pi}})}{\mathcal{E}(\sqrt{\frac{d\mu_i}{d\pi}})} - 1$.

Acceleration-accuracy trade-off

Larger correction factor[a] F
Larger acceleration, **lower** accuracy

Larger batch size n
Larger acceleration, **slower** evaluation

[a] Where it is defined: $\hat{S}_1 = \exp\left(\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(1)}_{k+1}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}^{(2)}_{k+1}) - \frac{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\hat{\sigma}^2_{m+1}}{F}\right)\right)$

Acceleration-accuracy trade-off

Larger correction factor[a] F

Larger acceleration, lower accuracy

Larger batch size n

Larger acceleration, **slower** evaluation

---

[a]Where it is defined: $\hat{S}_1 = \exp\left(\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) - \frac{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\hat{\sigma}_{m+1}^2}{F}\right)\right)$

Acceleration-accuracy trade-off

Larger correction factor[a] F

Larger acceleration, **lower** accuracy

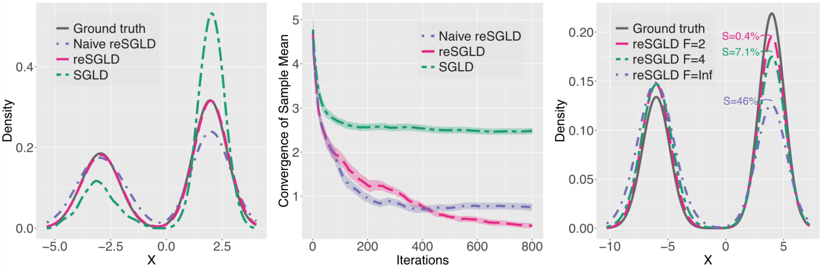
Larger batch size n

Larger acceleration, **slower** evaluation

[a] Where it is defined: $\hat{S}_1 = \exp\left(\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) - \frac{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\hat{\sigma}_{m+1}^2}{F}\right)\right)$

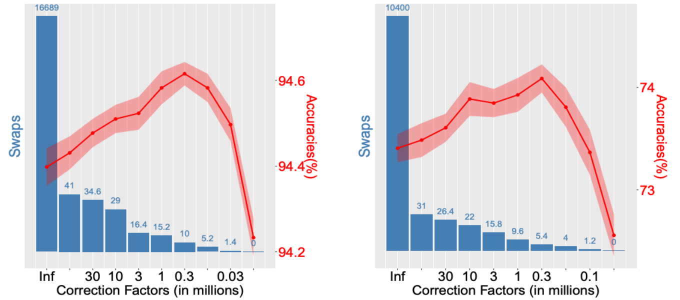
Experiments

Sampling from Gaussian mixture distributions



(a) $\widetilde{U}_1(x) \sim \mathcal{N}(U_1(x), 2^2)$. (b) Convergence evaluation. (c) $\widetilde{U}_3(x) \sim U_3(x) + 7t(\nu=10)$.

Figure 3: Evaluation of reSGLD on Gaussian mixture distributions, where reSGLD proposes to adaptively estimate the unknown corrections and the naïve reSGLD doesn't make any corrections to adjust the swapping rates.



(c) Effects of correction factors on CIFAR10

(d) Effects of correction factors on CIFAR100

Figure 4: More swaps don't necessarily lead to better performance.

Supervised Learning (II): Batch size matters

Table 1: Prediction accuracies (%) with different batch sizes on CIFAR10 & CIFAR100 using ResNet-20.

Batch	M-SGD	SGHMC	reSGHMC
		CIFAR10	
256	94.21±0.16	94.22±0.12	94.62±0.18
1024	94.49±0.12	94.57±0.14	**95.01±0.16**
		CIFAR100	
256	72.45±0.20	72.49±0.18	74.14±0.22
1024	73.31±0.18	73.23±0.20	**75.11±0.26**

Bayesian GAN for Semi-supervised Learning

Table 2: SEMI-SUPERVISED LEARNING ON CIFAR100 AND SVHN BASED ON DIFFERENT NUMBER OF LABELS.

N_s	CIFAR100		SVHN	
	SGHMC	reSGHMC	SGHMC	reSGHMC
2000	50.76±0.71	**55.53± 0.64**	88.75±0.44	**91.59±0.38**
3000	53.07±0.71	**57.09± 0.77**	91.32±0.41	**94.03±0.36**
4000	57.05±0.59	**62.23± 0.69**	91.92±0.41	**94.25±0.31**
5000	59.34±0.64	**64.83± 0.72**	92.63±0.46	**94.33±0.34**

Conclusion

Summary

Achieved

Algorithm
Scalable and adaptive.

Theory
The accelerated convergence implies an acceleration -accuracy trade-off

Experiments
Extensive experiments with significant improvements.

Future works

Generalization

Relax normal to the heavy-tailed generalization of Lévy-stable distribution [Şimşekli et al., 2019]

Variance reduction

Variance reduction [Xu et al., 2018] to obtain a larger acceleration effect.

Cattiaux, P., Guillin, A., and Wu, L.-M. (2010).
A Note on Talagrand's Transportation Inequality and Logarithmic Sobolev Inequality.
Prob. Theory and Rel. Fields, 148:285–334.

Chen, Y., Chen, J., Dong, J., Peng, J., and Wang, Z. (2019).
Accelerating Nonconvex Learning via Replica Exchange Langevin Diffusion.
In *Proc. of the International Conference on Learning Representation (ICLR)*.

Şimşekli, U., Sagun, L., and Gürbüzbalaban, M. (2019).
A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks.
In *Proc. of the International Conference on Machine Learning (ICML)*.

Kirkpatrick, S., Jr, D. G., and Vecchi, M. P. (1983).
Optimization by Simulated Annealing.
Science, 220(4598):671–680.

Marinari, E. and Parisi, G. (1992).
Simulated Tempering: A New Monte Carlo Scheme.
Europhysics Letters (EPL), 19(6):451–458.

Swendsen, R. H. and Wang, J.-S. (1986).
Replica Monte Carlo Simulation of Spin-Glasses.
Phys. Rev. Lett., 57:2607–2609.

Teh, Y. W., Thiéry, A., and Vollmer, S. (2016).
Consistency and Fluctuations for Stochastic Gradient Langevin Dynamics.
Journal of Machine Learning Research, 17:1–33.

Welling, M. and Teh, Y. W. (2011).
Bayesian Learning via Stochastic Gradient Langevin Dynamics.
In *Proc. of the International Conference on Machine Learning (ICML)*, pages 681–688.

Xu, P., Chen, J., Zou, D., and Gu, Q. (2018).
Global Convergence of Langevin Dynamics Based Algorithms for Nonconvex Optimization.
In *Proc. of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Zhang, Y., Liang, P., and Charikar, M. (2017).
A Hitting Time Analysis of Stochastic Gradient Langevin Dynamics.
In *Proc. of Conference on Learning Theory (COLT)*, pages 1980–2022.