Markov-Chain Monte Carlo Score Estimators for Variational Inference with Score Climbing

Anonymous Author
Anonymous Institution

Abstract

Recently, variational inference methods that minimize the inclusive Kullback-Leibler (KL) divergence using Markov-chain Monte Carlo (MCMC) have been developed. These methods perform stochastic gradient descent by obtaining noisy estimates of the score function using MCMC. In this paper, we compare three different ways to operate Markov-chains for VI, and compare the performance of different schemes. In particular, we propose the parallel state estimator, which averages a single state of multiple parallel Markov-chains. Compared to previously used MCMC based score climbing schemes, this estimator has lower variance enabling faster convergence. Our experiments show that, when using our proposed scheme, inclusive KL divergence minimization is competitive against evidence lower bound minimization.

1 Introduction

Given an observed data $\mathbf{x}$ and a latent variable $\mathbf{z}$, Bayesian inference aims to analyze $p(\mathbf{z}\,|\,\mathbf{x})$ given an unnormalized joint density $p(\mathbf{z}, \mathbf{x})$ where the relationship is given by Bayes' rule such that $p(\mathbf{z}\,|\,\mathbf{x}) = p(\mathbf{z}, \mathbf{x})/p(\mathbf{x}) \propto p(\mathbf{z}, \mathbf{x})$. Instead of working directly with the target distribution $p(\mathbf{z}\,|\,\mathbf{x})$, variational inference (VI, Jordan et al. 1999; Blei et al. 2017; Zhang et al. 2019) searches for a variational approximation $q_\lambda(\mathbf{z})$ that is similar to $p(\mathbf{z}\,|\,\mathbf{x})$ according to a discrepancy measure $D(p, q_\lambda)$.

Naturally, choosing a good discrepancy measure, or objective function, is critical to the problem. This fact had lead to a quest for suitable divergence measures (Salimans et al., 2015; Li and Turner, 2016; Dieng et al., 2017; Wang et al., 2018; Ruiz and Titsias, 2019). So far, the exclusive KL divergence $D_{\mathrm{KL}}(q_\lambda \parallel p)$ (or reverse KL divergence) has been used "exclusively" among various discrepancy measures. This is partly because the exclusive KL is defined as an average over $q_\lambda(\mathbf{z})$, which can be estimated efficiently. By contrast, the inclusive KL is defined as

$$D_{\mathrm{KL}}(p \parallel q_\lambda) = \int p(\mathbf{z} \mid \mathbf{x}) \log \frac{p(\mathbf{z} \mid \mathbf{x})}{q_\lambda(\mathbf{z})}\, d\mathbf{z} \tag{1}$$

$$= \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{z} \mid \mathbf{x})}{q_\lambda(\mathbf{z})} \right] \tag{2}$$

where the average is taken over $p(\mathbf{z}\,|\,\mathbf{x})$. Interestingly, this is a chicken-and-egg problem as our goal is to obtain $p(\mathbf{z}\,|\,\mathbf{x})$ in the first place. Despite this challenge, minimizing (2) has drawn the attention of researchers because it is believed to result in favorable statistical properties (Minka, 2005; MacKay, 2001).

For minimizing the inclusive KL divergence, Naesseth et al. (2020) and Ou and Song (2020) have recently proposed methods that perform stochastic gradient descent (SGD, Robbins and Monro 1951) with the score function estimated using Markov-chain Monte Carlo (MCMC). These MCMC score climbing schemes operate a Markov-chain in conjunction with the VI optimizer. In addition, within the MCMC kernel, they both use Metropolis-Hastings proposals generated from the variational approximation $\mathbf{z}^* \sim q_\lambda(\cdot)$. The MCMC kernel itself benefits from VI, enjoying better proposals over time without the need for computationally expensive proposals as in Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2011b; Betancourt, 2017). Also, in terms of computational cost, score climbing is efficient compared to other divergences since we do not need to differentiate through the likelihood.

While the methods by Naesseth et al. (2020) and Ou and Song (2020) are conceptually similar, they both use their MCMC kernels in different ways. At each SGD iteration, for estimating the score function, Naesseth et al. use a single sample generated from a relatively expansive

Preliminary work. Under review by AISTATS 2022. Do not distribute.

MCMC kernel, while Ou and Song average multiple samples generated from a cheaper MCMC kernel. We call the former scheme the *single state estimator* and the later the *sequential state estimator*. Given the two options, it is natural to ask, "which is better? An estimator with multiple cheap samples? or one with a single expensive sample?".

In this paper, we propose a third novel scheme, *the parallel state estimator*. The parallel state estimator operates N parallel Markov-chains parallel, where only a single state transition is performed on each chain. The variance of this estimator linearly decreases with the computational budget N, unlike the single and sequential state estimators. While the substantial variance reduction comes at the cost of slightly higher bias, we observe that this is not a significant downside in practice.

We compare the bias and variance of the three different schemes and conduct experiments on general Bayesian inference problems. Our results show that, given a similar computational budget N, the parallel state estimator results in the best performance. Also, score climbing VI with the parallel state estimator is competitive against evidence lower-bound (ELBO) maximization. Interestingly, this observation is against the conclusions of Dhaka et al. (2021) We further discuss this discrepancy in the Discussions Section 6.

Contribution Summary

- ➤ We propose the parallel state estimator for estimating the score function using MCMC (**Section 3.1**).
- ➤ We discuss the bias and variance of the MCMC score estimation schemes (**Section 3.2**).
- ➤ We experimentally compare the VI performance of the considered MCMC estimation schemes on general Bayesian inference benchmarks (**Section 4**).
- ➤ We also show that inclusive KL minimization with score climbing is competitive against exclusive KL divergence minimization (**Section 4**).

2 Background

2.1 Inclusive Variational Inference Until Now

Score Function and Variational Inference A typical way to perform VI is to use stochastic gradient descent (SGD, Robbins and Monro 1951; Bottou 1999), provided that unbiased gradient estimates of the optimization target $g(\boldsymbol{\lambda})$ are available. In this case, SGD is performed by repeating the update

$$\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g\left(\boldsymbol{\lambda}_{t-1}\right) \tag{3}$$

where $\gamma_1, \ldots, \gamma_T$ is a step-size schedule following the conditions of Robbins and Monro (1951); Bottou (1999). In the case of inclusive KL divergence minimization, obtaining g corresponds to estimating

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}} D_{\mathrm{KL}}(p \parallel q_{\lambda}) &= \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[-\nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})\right] && (4)\\ &= -\mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[s(\mathbf{z};\boldsymbol{\lambda})\right] && (5)\\ &\approx g(\boldsymbol{\lambda}) && (6) \end{aligned}$$

where $s(\mathbf{z};\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$ is known as the *score function*. Evidently, estimating $\nabla_{\boldsymbol{\lambda}} D_{\mathrm{KL}}(p \parallel q_{\lambda})$ requires integrating the score function over $p(\mathbf{z} \mid \mathbf{x})$, which is prohibitive. Different inclusive variational inference methods form a different estimator g.

Importance Sampling When it is easy to sample from the variational approximation $q_{\lambda}(\mathbf{z})$, one can use importance sampling (IS, Robert and Casella 2004; Owen 2013) for estimating g since

$$\begin{aligned} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[s(\mathbf{z};\boldsymbol{\lambda})\right] &\propto \mathbb{E}_{q_{\lambda}}\left[w(\mathbf{z})\, s(\mathbf{z};\boldsymbol{\lambda})\right] && (7)\\ &\approx \frac{1}{N}\sum_{i=1}^{N} w(\mathbf{z}^{(i)})\, s(\mathbf{z}^{(i)};\boldsymbol{\lambda}) && (8)\\ &= g_{IS}(\boldsymbol{\lambda}) && (9) \end{aligned}$$

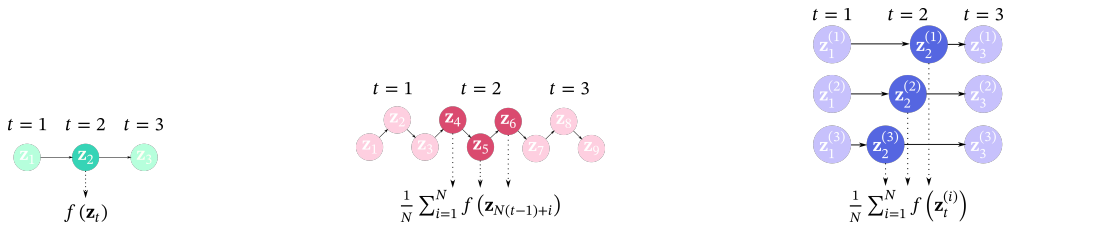
where $w(\mathbf{z}) = p(\mathbf{z},\mathbf{x})/q_{\boldsymbol{\lambda}}(\mathbf{z})$ is known as the *importance weight*, and $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$ are N independent samples from $q_{\boldsymbol{\lambda}}(\mathbf{z})$. This scheme is equivalent to adaptive IS methods (Cappé et al., 2008; Bugallo et al., 2017) since the IS proposal $q_{\boldsymbol{\lambda}}(\mathbf{z})$ is iteratively optimized based on the current samples. Although IS is unbiased, it is highly unstable in practice. A more stable alternative is to use the *normalized weight* $\widetilde{w}^{(i)} = w(\mathbf{z}^{(i)})/\sum_{i=1}^{N} w(\mathbf{z}^{(i)})$, which results in the self-normalized IS (SNIS) approximation. Unfortunately, SNIS still fails to converge even on moderate dimensional objectives and unlike IS, it is no longer unbiased (Robert and Casella, 2004; Owen, 2013).

3 Markov-chain Monte Carlo Estimators for Score Climbing

3.1 Overview of Estimation Strategies

Score Climbing with MCMC Recently, Naesseth et al. (2020) and Ou and Song (2020) proposed two similar but independent score climbing method that minimize the inclusive KL with SGD. Both methods estimate the score function gradient by operating a Markov-chain in parallel with the VI optimization sequence. They notably use MCMC kernels that can effectively utilize the variational approximation $q_{\boldsymbol{\lambda}_t}(\mathbf{z})$. Because of this, both methods are computationally more efficient than previous VI approaches (Ruiz and Titsias, 2019; Hoffman, 2017) that used expensive MCMC kernels such as Hamiltonian Monte Carlo.

Single State Estimator In Markovian score climbing (MSC), Naesseth et al. (2020) estimate the score gradient by performing an MCMC transition and estimate the



(a) Single State Estimator (b) Sequential State Estimator (c) Parallel State Estimator (proposed)

Figure 1: Visualization of different ways of combining MCMC with stochastic approximation variational inference. The index t denotes the stochastic approximation iteration. The dark circles represent the MCMC samples used for estimating the score gradient at $t = 2$.

Table 1: Computational Costs of Markov-chain Schemes

	Posterior Sampling			Stochastic gradient	
	$p(\mathbf{z}, \mathbf{x})$ # Eval.	$q_{\boldsymbol{\lambda}}(\mathbf{z})$ # Eval.	$q_{\boldsymbol{\lambda}}(\mathbf{z})$ # Samples	$p(\mathbf{z}, \mathbf{x})$ # Grad.	$q_{\boldsymbol{\lambda}}(\mathbf{z})$ # Grad.
Evidence Lower Bound Path Derivative	0	0	N	N	N
Single State Estimator with CIS Kernel (single-CIS)	$N-1$	N	$N-1$	0	1^1 or N^2
Sequential State Estimator with IMH Kernel (seq.-IMH)	N	$N+1$	N	0	N
Parallel State Estimator with IMH Kernel (par.-IMH)	N	$2N$	N	0	N

* We assume that the parameters are cached as much as possible.
* N is the number of samples used in each method.
[1] Vanilla CIS kernel.
[2] Rao-Blackwellized CIS kernel.

score function gradient as

$$\mathbf{z}_t \sim K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_{t-1}, \cdot)$$
$$g_{\text{single-CIS}}(\boldsymbol{\lambda}) = s(\mathbf{z}_t; \boldsymbol{\lambda})$$

where $K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_{t-1}, \cdot)$ is a MCMC kernel leaving $p(\mathbf{z} \mid \mathbf{x})$ invariant and $g_{\text{single}}(\boldsymbol{\lambda})$ denotes the score estimator. For the MCMC kernel, they propose a new type of kernel inspired by particle MCMC (Andrieu et al., 2010, 2018), the conditional importance sampling (CIS) kernel. Since the estimator uses *a single state* created by the CIS kernel, we call it the single state estimator with the CIS kernel (single-CIS). The CIS kernel internally uses N samples from the $q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z})$, hence the dependence on $\boldsymbol{\lambda}_{t-1}$. When compared to MCMC kernels that only use a single sample from $q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z})$, it is N times more expensive, but hopefully, statistically superior.

Sequential State Estimator On the other hand, at each SGD iteration t, Ou and Song (2020) perform N sequential Markov-chain transitions and use the average of the intermediate states for estimation. That is, for the index $i \in \{1, \ldots, N\}$,

$$\mathbf{z}_{T+i} \sim K^i_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_T, \cdot)$$
$$g_{\text{seq.-IMH}}(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{i=1}^{N} s(\mathbf{z}_{T+i}; \boldsymbol{\lambda})$$

where $\mathbf{z}_T$ is the last Markov-chain state of the previous SGD iteration $t-1$. $K^i_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_T, \cdot)$ denotes the MCMC kernel sequentially applied i times. For the MCMC kernel, they use the independent Metropolis-Hastings (IMH, Robert and Casella 2004, Algorithm 25 Hastings 1970) algorithm, which uses only a single sample from $q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z})$ (notice the dependence on $\boldsymbol{\lambda}_{t-1}$). Therefore, the cost of N state transitions with IMH is similar to a single transition with CIS. Since the estimator uses sequential states, we call it the sequential state estimator with the IMH kernel (seq.-IMH)

Overview The single and sequential state estimators represent two different ways of using a fixed computational budget. The former uses a single sample generated expensively, while the latter uses multiple samples generated cheaply. Illustrations of the two schemes are provided in Figures 1a and 1b.

Parallel State Estimator In this work, we propose a new scheme into the mix: *the parallel state estimator*. Like the sequential state estimator, we use the cheaper IMH kernel, but instead of applying the MCMC kernel N times to a single chain, we apply the MCMC kernel a single time to N *parallel Markov-chains*. That is, for each Markov-chain indexed by $i \in \{1, \ldots, N\}$,

$$\mathbf{z}_t^{(i)} \sim K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_{t-1}^{(i)}, \cdot)$$
$$g_{\text{par.-IMH}}(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{i=1}^{N} s(\mathbf{z}_t^{(i)}; \boldsymbol{\lambda})$$

where $\mathbf{z}_{t-1}^{(i)}$ is the state of the ith chain at the previous SGD step. Computationally speaking, we are still applying $K(\mathbf{z}_{t-1}^{(i)})$ N times in total, so the cost is similar to the sequential state estimator. However, the Markov-chains are N times shorter, which, in a traditional MCMC view, might seem to result in worse statistical performance. An illustration of the parallel state estimator is shown in Figure 1c.

Computational Cost The three schemes using the CIS kernel and the IMH kernel can have different computational costs depending on the parameter N. The computational costs of each scheme are organized in Table 1 while detailed pseudocodes of the considered schemes are provided in the *supplementary material*. In the CIS kernel, N controls the number of internal proposals sampled from $q_{\boldsymbol{\lambda}}(\mathbf{z})$. In the sequential and parallel state estimators, the IMH kernel only uses a single sample from $q_{\boldsymbol{\lambda}}(\mathbf{z})$, but applies the kernel N times. Assuming caching is done as much as possible, the parallel state estimator needs twice the density evaluations of $q_{\boldsymbol{\lambda}}(\mathbf{z})$ compared to other methods. However, this added cost is minimal since the overall computational cost is dominated by $p(\mathbf{z}, \mathbf{x})$. When estimating the score, the single state estimator computes $\nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$ only once, while for the sequential and parallel state estimators compute it N times. However, Naesseth et al. (2020) also discuss a Rao-Blackwellized version of the CIS kernel, which also computes the gradient N times. Lastly, notice that score climbing does not need to differentiate through the likelihood, unlike ELBO maximization, making it's base computational cost significantly cheaper.

3.2 Theoretical Analysis of Bias

Adaptive MCMC and Ergodicity For bounded functions, an upper bound on the bias of MCMC estimators can be easily derived from the convergence rates of MCMC kernels. In the context of score climbing VI, the MCMC convergence is a subtle subject since the kernel is now *adaptive* as it depends on $\boldsymbol{\lambda}_t$, which depends on all of the past MCMC samples. This is clearly the type of problem adaptive MCMC algorithms have been concerned with (Andrieu and Moulines, 2006). Fortunately, our setting crucially differs from adaptive MCMC in that we do not seek to obtain asymptotically unbiased samples. Instead, we only use the MCMC samples acquired during each SGD step, within the corresponding SGD step, where $\boldsymbol{\lambda}_t$ is fixed. That is, our MCMC kernel is instantaneously not adaptive, and we are thus free to use the corresponding ergodic convergence rates. In addition, as far as Deoblin's condition holds such that $w^* = \sup_{\mathbf{z},\boldsymbol{\lambda}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}}(\mathbf{z}) < \infty$ for all $\boldsymbol{\lambda}_t$ and the SGD step-size sequence satisfies the diminishing adaptation condition (Roberts and Rosenthal, 2007), the kernel will eventually result in asymptotically unbiased samples.

Boundedness Assumption We generally assume that the score function is bounded such that, $\| \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z}) \| < L$ for any $\boldsymbol{\lambda}$. This boundedness assumption is reasonable since theoretical guarantees of SGD often assume Lipschitz-continuity.

Theorem 1. *(Bias of seq.-IMH) Assuming* $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}_t}(\mathbf{z}) < \infty$ *and the score function is bounded such that* $| s(\mathbf{z}; \boldsymbol{\lambda}) | \leq \frac{L}{2}$*, the bias of the sequential state estimator with an IMH kernel averaging N states at iteration t is bounded as*

$$\text{Bias}\left[g_{\text{seq.},t}\right] \leq L\left(1 - \frac{N}{2\,w^*}\right) + \mathcal{O}\left(\left(\frac{1}{w^*}\right)^2\right)$$

Proof. The proof is in the *supplementary material*.

Theorem 2. *(Bias of par.-IMH) Assuming* $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}_\tau}(\mathbf{z}) < \infty$ *and that the score function is bounded as* $| s(\mathbf{z}; \boldsymbol{\lambda}) | \leq \frac{L}{2}$*, the bias of the parallel state estimator with an IMH kernel and N parallel chains at iteration t is bounded as*

$$\text{Bias}\left[g_{\text{par.},t}\right] \leq L\left(1 - \frac{1}{w^*}\right).$$

Proof. The proof is in the *supplementary material*.

For the single-CIS estimator, our proof is based on the fact that the CIS kernel is identical to the iterated sampling importance resampling (i-SIR) algorithm by Andrieu et al. (2018). We also note that the CIS kernel can be reformulated as a multiple-try Martino (2018, Table 12) accept-reject type kernel that uses Barker's acceptance function (Barker, 1965), resulting in the ensemble MCMC sampler Austad (2007); Neal (2011a).

Theorem 3. *(Bias of single-CIS) For a CIS kernel with N internal proposals, assuming* $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}}(\mathbf{z}) < \infty$ *for* $\forall \boldsymbol{\lambda}$*,* $N > 2$*, and that the score function is bounded such that* $| s(\mathbf{z}; \boldsymbol{\lambda}) | \leq \frac{L}{2}$*, the bias of the single state estimator at iteration t is bounded as*

$$\text{Bias}\left[g_{\text{cis.},t}\right] \leq L\left(1 - \frac{N-1}{2\,w^* + N - 2}\right)$$

Proof. The proof is in the *supplementary material*.

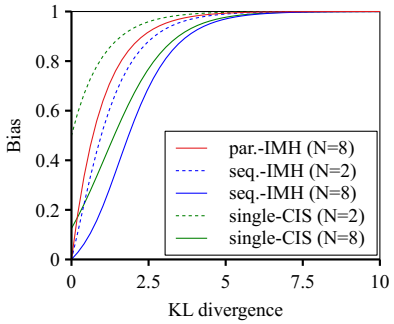


Figure 2: Relative bias of different estimators. For simplicity, we take $w^* = \exp(D_{\text{KL}}(p \parallel q))$.

Reducing Bias by Increasing N For the seq.-IMH estimator and single-CIS estimator, increasing N improves the bias decrease rate. However, the bounds depend on w^*, which is bounded below exponentially, as shown by the following proposition.

Proposition 1. *$w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_\lambda(\mathbf{z})$ is bounded below exponentially by the KL divergence such that*

$$\exp\left(D_{\mathrm{KL}}(p(\cdot \mid \mathbf{x}) \parallel q_\lambda(\cdot))\right) \leq w^*.$$

Proof. The proof is in the *supplementary material*.

Thus, in the initial steps of SGD, where the KL divergence is considerable, the bias will also be significant regardless of N. Therefore, increasing N will not bring a significant reduction in bias. To illustrate this point, we visualized the bounds in Figure 2.

3.3 Theoretical Analysis of Variance

For MCMC estimators, variance often dominates the mean squared error. Therefore, variance gives a better sense of their practical performance.

Variance of Single State Estimator The variance of the single estimator is given by the law of total variance such that

$$\begin{aligned}
&\mathbb{V}\left[g_{\text{single}} \right] \\
&= \mathbb{E}\left[\mathbb{V}_{K_{\lambda_{t-1}}(\mathbf{z}_{t-1},\mathbf{z})}\left[s(\boldsymbol{\lambda};\mathbf{z}) \mid \mathbf{z}_{t-1} \right] \right] \\
&\quad + \mathbb{V}\left[\mathbb{E}_{K_{\lambda_{t-1}}(\mathbf{z}_{t-1},\mathbf{z})}\left[s(\boldsymbol{\lambda};\mathbf{z}) \mid \mathbf{z}_{t-1} \right] \right], \qquad (10)
\end{aligned}$$

and assuming stationarity such that $\mathbf{z}_{t-1} \sim p(\mathbf{z} \mid \mathbf{x})$,

$$\begin{aligned}
&= \mathbb{E}_{p(\cdot|\mathbf{x})}\left[\mathbb{V}_{K_{\lambda_{t-1}}(\mathbf{z}_{t-1},\mathbf{z})}\left[s(\boldsymbol{\lambda};\mathbf{z}) \mid \mathbf{z}_{t-1} \right] \right] \\
&\quad + \mathbb{V}_{p(\cdot|\mathbf{x})}\left[\mathbb{E}_{K_{\lambda_{t-1}}(\mathbf{z}_{t-1},\mathbf{z})}\left[s(\boldsymbol{\lambda};\mathbf{z}) \mid \mathbf{z}_{t-1} \right] \right] \qquad (11) \\
&= \mathbb{V}_{p(\cdot|\mathbf{x})}\left[g_{\text{single}} \right] \qquad (12) \\
&= \sigma^2. \qquad (13)
\end{aligned}$$

Ideally, the variance will be equal to the variance of an independent draw from the posterior. This also suggests that, under stationarity, the variance of a single state estimator will be similar with any MCMC kernel.

Variance of Parallel State Estimator On the other hand, the parallel state estimator can be seen as an average of *i.i.d.* single state estimators. Therefore, under stationarity, the variance is

$$\mathbb{V}\left[g_{\text{par.}} \right] = \frac{\sigma^2}{N} \qquad (14)$$

Note that the variance reduction rate does not necessarily require stationarity. The parallel state estimator thus *always* enjoys $\mathcal{O}(1/N)$ variance reduction.

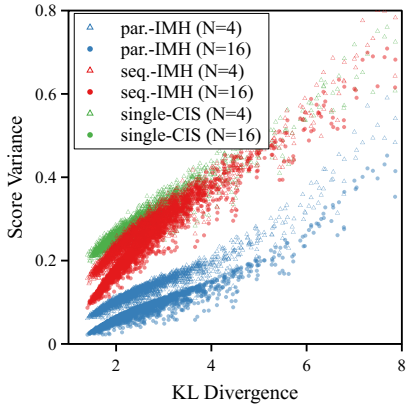


Figure 3: Variance of score function estimated using the three estimators depending on N and the KL divergence. Each point represent a random $\boldsymbol{\lambda}$ where the score is evaluated.

Variance of Sequential State Estimator Assuming stationarity, the variance of the sequential state estimator is given as

$$\mathbb{V}\left[g_{\text{seq.}} \right] = \frac{\sigma^2}{N} + \frac{2}{N^2} \sum_{i<j}^{N} \mathbb{E}_{p(\mathbf{z}_T|\mathbf{x})}\left[\mathrm{Cov}\left(\mathbf{z}_i, \mathbf{z}_j \mid \mathbf{z}_T \right) \right]. \qquad (15)$$

which is similar to the variance of a regular MCMC estimator. Here, $T = N(t-1)$ is the last state of the chain at the previous SGD iteration $t-1$. (Detailed derivation is in the *supplementary material*.) If the covariance term does not exist, the performance will be equal to the parallel state estimator. However, due to rejections in the MCMC kernel, adjacent states will have some level of positive covariance. Unfortunately, the rejection rate will be high when the KL divergence is significant, as implied by Proposition 1. Thus, in practice, we can expect the performance of the sequential state estimator to be worse than the parallel state estimator. This is the case as shown in the numerical simulation in Section 4.1.

4 Evaluations

4.1 Numerical Simulation

Experimental Setup We first present numerical simulation results of the three estimators. We chose the target posterior to be a 10 dimensional white Gaussian. We then randomly generated 2048 random $\boldsymbol{\lambda} = \boldsymbol{\mu}$ where $q_\lambda(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\mu}$ is drawn from a multivariate Student's T distribution, and $\boldsymbol{\Sigma} = 1.5^2\mathbf{I}$. Using the 2048 random $q_{\mu,\Sigma}$s, we simulate 128 Markov-chains of length $T = 50$ and estimate the bias and variance of the score function.

Table 2: Classification Accuracy and Log Predictive Density on Logistic Regression Problems

	`Pima Indians`		`heart disease`		`German credit`	
	Test Accuracy	Test LPD	Test Accuracy	Test LPD	Test Accuracy	Test LPD
ELBO	**0.77 (0.77, 0.77)**	-0.53 (-0.55, -0.51)	0.84 (0.84, 0.84)	**-0.40 (-0.40, -0.39)**	**0.77 (0.77, 0.77)**	-0.54 (-0.58, -0.50)
par.-IMH (ours)	**0.77 (0.77, 0.77)**	**-0.51 (-0.51, -0.50)**	**0.85 (0.84, 0.85)**	**-0.40 (-0.40, -0.40)**	**0.77 (0.77, 0.77)**	**-0.50 (-0.50, -0.50)**
seq.-IMH	0.67 (0.65, 0.69)	-0.71 (-0.76, -0.65)	0.79 (0.78, 0.80)	-0.45 (-0.46, -0.44)	0.76 (0.76, 0.76)	-0.51 (-0.51, -0.51)
single-CIS	0.69 (0.67, 0.71)	-0.68 (-0.73, -0.62)	0.79 (0.78, 0.80)	-0.46 (-0.48, -0.44)	0.76 (0.75, 0.76)	-0.51 (-0.52, -0.51)
single-CISRB	0.71 (0.69, 0.72)	-0.62 (-0.67, -0.58)	0.80 (0.79, 0.81)	-0.44 (-0.45, -0.43)	0.76 (0.76, 0.76)	-0.52 (-0.52, -0.51)
single-HMC	0.75 (0.75, 0.75)	-0.52 (-0.53, -0.52)	0.80 (0.79, 0.81)	-0.45 (-0.46, -0.44)	**0.77 (0.76, 0.77)**	-0.61 (-0.72, -0.49)
SNIS	0.72 (0.71, 0.72)	-0.59 (-0.61, -0.57)	0.78 (0.77, 0.79)	-0.46 (-0.48, -0.45)	0.75 (0.75, 0.76)	-0.52 (-0.52, -0.52)

* LPD denotes the average log predictive density.
* The numbers in the parentheses denote the 80% bootstrap confidence intervals computed from 100 repetitions.

Results The variance results are shown in Figure 3. We do not present the bias as all methods were visually indistinguishable. From the results, when the KL divergence is significant, we see that seq.-IMH and single-CIS do not benefit from increasing N. On the other hand, the parallel state estimator always benefits from increasing N. The variance of the parallel state estimator follows a linear trend until the large KL divergence prevents the chains from achieving stationary.

4.2 Baselines and Implementation

Implementation For the realistic experiments, we implemented score climbing VI on top of the Turing (Ge et al., 2018) probabilistic programming framework. Our implementation works with any model described in Turing, which automatically handles distributions with constrained support (Kucukelbir et al., 2017). We use the ADAM optimizer by Kingma and Ba (2015) with a learning rate of 0.01 in all of the experiments. The computational budget is set to $N = 10$ and $T = 10^4$ for all experiments unless specified.

We compare the following methods.

❶ **par.-IMH**: Score climbing with the parallel state estimator and the IMH kernel.
❷ **seq.-IMH**: Score climbing with the sequential state estimator and the IMH kernel
❸ **single-CIS**: Score climbing with the single state estimator and the CIS kernel (Naesseth et al., 2020).
❹ **single-CISRB**: Rao-Blackwellized version of single-CIS (Naesseth et al., 2020).
❸ **single-HMC**: Score climbing with the single state estimator and the HMC kernel.
❺ **SNIS**: adaptive IS using SNIS (as discussed in Section 2.1).
❻ **ELBO**: evidence lower-bound maximization with automatic differentiation VI (Ranganath et al., 2014; Kucukelbir et al., 2017) and the path derivative estimator (Roeder et al., 2017).

For ELBO, we use only a single sample as originally described by Roeder et al. (2017). This also ensures a fair comparison against inclusive KL minimization methods since the iteration complexity of computing the ELBO gradient can be easily a few orders of magnitude larger. Also, we only use single-HMC in the logistic regression experiment due to its high computational demands,

4.3 Hierarchical Logistic Regression

Experimental Setup We first perform logistic regression with the `Pima Indians` diabetes ($\mathbf{z} \in \mathbb{R}^{11}$, Smith et al. 1988), `German credit` ($\mathbf{z} \in \mathbb{R}^{27}$), and `heart disease` ($\mathbf{z} \in \mathbb{R}^{16}$, Detrano et al. 1989) datasets obtained from the UCI repository (Dua and Graff, 2017). 10% of the data points were randomly selected in each of the 100 repetitions as test data.

Probabilistic Model Instead of the usual single-level probit/logistic regression models, we choose a more complex hierarchical logistic regression model

$$\begin{aligned} \sigma_\beta, \sigma_\alpha &\sim \mathcal{N}^+(0, 1.0) \\ \boldsymbol{\beta} &\sim \mathcal{N}(\mathbf{0},\, \sigma_\beta^2 \mathbf{I}) \\ p &\sim \mathcal{N}(\mathbf{x}_i^\top \boldsymbol{\beta} + \alpha,\, \sigma_\alpha^2) \\ y_i &\sim \text{Bernoulli-Logit}\,(p) \end{aligned}$$

where $\mathcal{N}^+(\mu, \sigma)$ is a positive constrained normal distribution with mean μ and standard deviation σ, $\mathbf{x}_i$ and y_i are the feature vector and target variable of the ith datapoint. The extra degrees of freedom σ_β and σ_α make this model relatively more challenging.

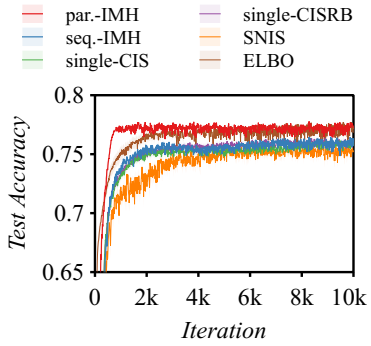
Results The test accuracy and test log predictive density (Test LPD) results are shown in Table 2. Our proposed parallel state estimator (par.-IMH) achieves the best accuracy and predictive density results. Despite having access to high-quality HMC samples, single-HMC shows poor performance. This supports our analysis that par.-IMH with $N > 1$ superior variance reduction to the

Table 3: Classification Accuracy and Log Predictive Density on Logistic Gaussian Process Problems

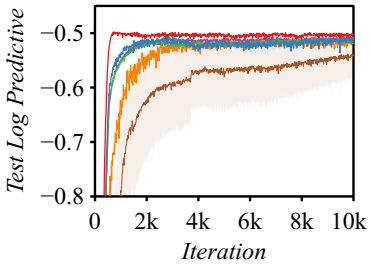
	`sonar`		`ionosphere`		`breast cancer`	
	Test Accuracy	Test LPD	Test Accuracy	Test LPD	Test Accuracy	Test LPD
ELBO	**0.86 (0.84, 0.88)**	-0.42 (-0.43, -0.40)	0.89 (0.88, 0.90)	-0.34 (-0.36, -0.33)		
par.-IMH	0.85 (0.83, 0.87)	**-0.37 (-0.39, -0.35)**	**0.92 (0.91, 0.93)**	**-0.30 (-0.32, -0.28)**	0.96 (0.96, 0.97)	**-0.16 (-0.17, -0.15)**
seq.-IMH	0.85 (0.83, 0.87)	-0.39 (-0.41, -0.38)	0.91 (0.89, 0.92)	-0.33 (-0.34, -0.31)	0.97 (0.96, 0.97)	-0.21 (-0.21, -0.20)
single-CIS	0.85 (0.83, 0.87)	-0.40 (-0.41, -0.38)	0.90 (0.89, 0.92)	-0.33 (-0.35, -0.32)	0.96 (0.96, 0.97)	-0.21 (-0.21, -0.20)
single-CISRB	0.85 (0.82, 0.87)	-0.39 (-0.40, -0.37)	0.90 (0.89, 0.92)	-0.32 (-0.33, -0.30)	0.96 (0.95, 0.97)	-0.20 (-0.20, -0.19)
SNIS	0.85 (0.83, 0.88)	-0.39 (-0.41, -0.37)	0.91 (0.89, 0.92)	-0.31 (-0.33, -0.30)	0.96 (0.95, 0.97)	-0.18 (-0.19, -0.18)

* LPD denotes the average log predictive density.
* The numbers in the parentheses denote the 80% bootstrap confidence intervals computed from 30 repetitions.

single state estimator. Also, seq.-IMH showed poor performance overall due to the correlated samples. Among the two CIS kernel-based methods, single-CISRB performs only marginally better than single-CIS.



(a) Test Accuracy



(b) Test Log Predictive Density

Figure 4: Test accuracy and log predictive density on the `german` dataset. The solid lines and colored regions are the mean and 80% bootstrap confidence interval computed from 100 repetitions.

Inclusive KL v.s. Exclusive KL While both ELBO and par.-IMH showed similar numerical performance, they chose different optimization paths in the parameter space. This is shown in Figure 4. While the test accuracy suggests that ELBO converges quickly around $t = 2000$ (Figure 4a), in terms of uncertainty estimate, it takes much longer to converge (Figure 4b). This shows that inclusive KL minimization chooses a path that has better density coverage as expected.

4.4 Gaussian Process Classification

Experimental Setup For a more challenging problem, we perform classification with latent Gaussian processes (Rasmussen and Williams, 2006; Nguyen and Bonilla, 2014). The simplified probabilistic model is

$$\begin{aligned} \log \theta &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ f &\sim \mathcal{GP}(\mathbf{0}, \mathbf{\Sigma}_\theta) \\ y_i &\sim \text{Bernoulli-Logit}(f(\mathbf{x}_i)) \end{aligned}$$

where we chose a Matérn 5/2 covariance kernel with automatic relevance determination (Neal, 1996). For the datasets, we use the `sonar` ($\mathbf{z} \in \mathbb{R}^{249}$, Gorman and Sejnowski 1988), `ionosphere` ($\mathbf{z} \in \mathbb{R}^{351}$, Sigillito et al. 1989), and `breast cancer` ($\mathbf{z} \in \mathbb{R}^{523}$, Wolberg and Mangasarian 1990) datasets. For `breast`, we preprocessed the input features with z-standardization. 10% of the data points were randomly selected in each of the 100 repetitions as test data. For this experiment, the iteration complexity of ELBO is almost two orders of magnitude larger than all inclusive KL minimization methods.

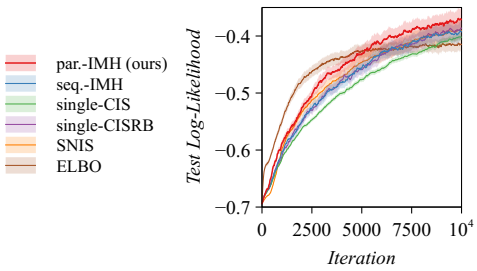
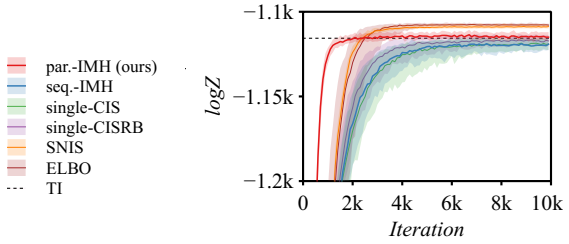


Figure 5: Test log predictive density on the `sonar` dataset. The solid lines and colored regions are the medians and 80% percentiles computed from 30 repetitions.

Result The results are shown in Table 3. Again, among inclusive KL minimization, the parallel state estimator (par.-IMH) achieved the best results. Compared to ELBO, its accuracy was lower on `sonar`, but the uncertainty estimates were much better. This is better shown in Figure 5, where ELBO quickly converges to a point with poor uncertainty calibration. Meanwhile, on `breast`, ELBO gives better uncertainty estimates than inclusive KL minimization methods. This happens when the modal estimate (preferred by the exclusive KL) gives good accuracy and uncertainty estimates.

4.5 Marginal Likelihood Estimation

Experimental Setup Lastly, we now estimate the marginal log-likelihood of a hierarchical regression model with partial pooling (`radon`, $\mathbf{z} \in \mathbb{R}^{175}$, Gelman and Hill 2007) for modeling radon levels in U.S homes. `radon` contains multiple posterior degeneracies from the hierarchy. We estimated the reference marginal likelihood using *thermodynamic integration* (TI, Gelman and Meng 1998; Neal 2001; Lartillot and Philippe 2006) with HMC implemented by Stan (Carpenter et al., 2017; Betancourt, 2017).



(a) `radon`

Figure 6: Marginal log-likelihood estimates on the `radon` dataset. The solid lines and colored regions are the medians and 80% percentiles computed from 100 repetitions.

Results The results are shown in Figure 6. par.-IMH converges quickly and provides the most accurate estimate. By contrast, other estimators converge much slowly. SNIS and ELBO, on the other hand, overestimate $\log Z$, which can be attributed to the mode-seeking behavior of ELBO and the small sample bias of SNIS.

5 Related Works

Inclusive KL minimization Our method directly builds on top of MSC (Naesseth et al., 2020), which minimizes the inclusive KL divergence. Concurrently, Ou and Song (2020) proposed JSA which they for training variational autoencoders with discrete latent variables. Unlike MSC, JSA can specifically be applied to models with *i.i.d.* data with minibatches. Before the two, only a few have proposed methods that minimize the inclusive KL using SGD. Notably, Bornschein and Bengio (2015) use SNIS for estimating the stochastic gradients, while Li et al. (2017) use an MCMC kernel to refine samples from $q_\lambda(\mathbf{z})$ to better resemble samples from $p(\mathbf{z} \mid \mathbf{x})$.

MCMC for VI Not restricted to inclusive KL minimization, MCMC has been widely utilized in VI. For example, Salimans et al. (2015); Ruiz and Titsias (2019) construct alternative divergence bounds from samples of an MCMC sampler.

Adaptive MCMC As pointed out by Ou and Song (2020), using q_λ within the MCMC kernel makes score climbing structurally equivalent to adaptive MCMC. In particular, Andrieu and Thoms (2008); Garthwaite et al. (2016) discuss the use of stochastic approximation in adaptive MCMC. Also, Andrieu and Moulines (2006); Keith et al. (2008); Holden et al. (2009); Giordani and Kohn (2010) specifically discuss adapting the propsosal of IMH kernels. Most similar to score climbing VI is the work of Keith et al. (2008) where they propose to use *cross-entropy minimization* (Barbakh et al., 2009), which is mathematically identical to inclusive VI. More recently, several other methods that apply variational inference for adapting the MCMC kernel have been developed. For adapting the proposals of an IMH sampler, Habib and Barber (2019) minimize the exclusive KL divergence while Neklyudov et al. (2019) minimize the symmetric KL divergence. And for HMC, Zhang et al. (2018); Campbell et al. (2021) have proposed to use score matching, ELBO maximization, and kernelized Stein discrepancy minimization.

6 Discussions

In this paper, we compared three different MCMC score estimators used for inclusive KL divergence minimization. Among the three estimators, the parallel state estimator that we proposed showed substantial variance reduction when increasing the number of samples. We demonstrated the performance of the parallel state estimator on general Bayesian inference tasks.

In our results, minimizing the inclusive KL divergence showed to be competitive against exclusive KL divergence minimization. This is not in line with the previous conclusions of Dhaka et al. (2021) that inclusive KL divergence does not work in high-dimensional problems (Dhaka et al. consider few hundreds of dimensions). While it is true that the inclusive KL fails in high dimensional and *correlated* posteriors, it is questionable how correlated posteriors really are in practice. Also, as shown in Figure 4 using the parallel estimator for score climbing results in vastly improved performance. This suggests that the negative results on realistic problems obtained by Dhaka et al. (2021) may be a problem of the inference algorithm rather than the inclusive KL itself. Our results motivate the development of inference algorithms for alternative divergence measures, including the inclusive KL.

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C., Lee, A., and Vihola, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2).

Andrieu, C. and Moulines, É. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, 16(3).

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Austad, H. M. (2007). *Parallel Multiple Proposal MCMC Algorithms*. Master thesis, Norwegian University of Science and Technology.

Barbakh, W. A., Wu, Y., and Fyfe, C. (2009). *Cross Entropy Methods*, volume 249, pages 151–174. Springer Berlin Heidelberg, Berlin, Heidelberg.

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton electron plasma. *Australian Journal of Physics*, 18(2):119.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434 [stat]*.

Betancourt, M. (2020). Hierarchical Modeling.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bornschein, J. and Bengio, Y. (2015). Reweighted wake-sleep. In *Proceedings of the 3rd International Conference on Learning Representation (ICLR)*, San Diego, California, USA.

Bottou, L. (1999). On-line learning and stochastic approximations. In *On-Line Learning in Neural Networks*, pages 9–42. Cambridge University Press, first edition.

Bugallo, M. F., Elvira, V., Martino, L., Luengo, D., Miguez, J., and Djuric, P. M. (2017). Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79.

Campbell, A., Chen, W., Stimper, V., Hernandez-Lobato, J. M., and Zhang, Y. (2021). A gradient based strategy for Hamiltonian Monte Carlo hyperparameter optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1238–1248. PMLR.

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1).

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., Guppy, K. H., Lee, S., and Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310.

Dhaka, A. K., Catalina, A., Welandawe, M., Andersen, M. R., Huggins, J., and Vehtari, A. (2021). Challenges and opportunities in high-dimensional variational inference. *arXiv:2103.01085 [cs, stat]*.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. (2017). Variational inference via χ upper bound minimization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, pages 2729–2738, Long Beach, California, USA. Curran Associates, Inc.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Garthwaite, P. H., Fan, Y., and Sisson, S. A. (2016). Adaptive optimal scaling of Metropolis–Hastings algorithms using the Robbins–Monro process. *Communications in Statistics - Theory and Methods*, 45(17):5098–5111.

Ge, H., Xu, K., and Ghahramani, Z. (2018). Turing: A language for flexible probabilistic inference. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pages 1682–1690. ML Research Press.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, Cambridge; New York.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2).

Giordani, P. and Kohn, R. (2010). Adaptive independent Metropolis–Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics*, 19(2):243–259.

Gorman, R. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1(1):75–89.

Habib, R. and Barber, D. (2019). Auxiliary variational MCMC. In *International Conference on Learning Representations*.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hoffman, M. D. (2017). Learning deep latent Gaussian models with Markov chain Monte Carlo. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1510–1519. PMLR.

Holden, L., Hauge, R., and Holden, M. (2009). Adaptive independent Metropolis–Hastings. *The Annals of Applied Probability*, 19(1).

Jiang, Y. H., Liu, T., Lou, Z., Rosenthal, J. S., Shangguan, S., Wang, F., and Wu, Z. (2021). MCMC confidence intervals and biases. *arXiv:2012.02816 [math, stat]*.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Keith, J. M., Kroese, D. P., and Sofronov, G. Y. (2008). Adaptive independence samplers. *Statistics and Computing*, 18(4):409–420.

Kim, H., Lee, J., and Yang, H. (2021). Adaptive strategy for resetting a non-stationary markov chain during learning via joint stochastic approximation. In *Proceedings of the 3rd Symposium on Advances in Approximate Bayesian, to Appear*.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, San Diego, California, USA.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45.

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207.

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29. Curran Associates, Inc.

Li, Y., Turner, R. E., and Liu, Q. (2017). Approximate inference with amortised MCMC. *arXiv:1702.08343 [cs, stat]*.

MacKay, D. J. (2001). Local minima, symmetry-breaking, and model pruning in variational free energy minimization. Technical Report.

Martino, L. (2018). A review of multiple try MCMC algorithms for signal processing. *Digital Signal Processing*, 75:134–152.

Mengersen, K. L. and Tweedie, R. L. (1996). Rates of convergence of the hastings and metropolis algorithms. *The Annals of Statistics*, 24(1):101–121.

Minka, T. (2005). Divergence measures and message passing. Technical Report MSR-TR-2005-173.

Naesseth, C., Lindsten, F., and Blei, D. (2020). Markovian score climbing: Variational inference with KL(p||q). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 15499–15510. Curran Associates, Inc.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer New York, New York, NY.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Neal, R. M. (2011a). MCMC Using Ensembles of States for Problems with Fast and Slow Variables such as Gaussian Process Regression. Dept. of Statistics Technical Report 1011, University of Toronto.

Neal, R. M. (2011b). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, Handbooks of Modern Statistical Methods, pages 113–162. Chapman and Hall/CRC, first edition.

Neklyudov, K., Egorov, E., Shvechikov, P., and Vetrov, D. (2019). Metropolis-Hastings view on variational inference and adversarial training. *arXiv:1810.07151 [cs, stat]*.

Nguyen, T. V. and Bonilla, E. V. (2014). Automated variational inference for Gaussian process models. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ou, Z. and Song, Y. (2020). Joint stochastic approximation and its application to learning discrete latent variable models. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 929–938. ML Research Press.

Owen, A. B. (2013). *Monte Carlo Theory, Methods and Examples*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. ML Research Press.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Comput. Mach. Learn. MIT Press, Cambridge, Mass.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of Applied Probability*, 44(2):458–475.

Roeder, G., Wu, Y., and Duvenaud, D. K. (2017). Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ruiz, F. and Titsias, M. (2019). A contrastive divergence for combining variational inference and MCMC. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 5537–5545. ML Research Press.

Salimans, T., Kingma, D., and Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1218–1226, Lille, France. PMLR.

Sigillito, V. G., Wing, S., Hutton, L. V., and Baker, K. L. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10:262–266.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265.

Wang, D., Liu, H., and Liu, Q. (2018). Variational inference with tail-adaptive f-Divergence. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31. Curran Associates, Inc.

Wolberg, W. H. and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23):9193–9196.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2019). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Zhang, C., Shahbaba, B., and Zhao, H. (2018). Variational Hamiltonian Monte Carlo via score matching. *Bayesian Analysis*, 13(2).

Markov-Chain Monte Carlo Score Estimators for Variational Inference with Score Climbing
Appendix

A Computational Resources

All of our experiments presented in this paper were executed on a server with 20 Intel Xeon E5–2640 CPUs and 64GB RAM. Each of the CPUs has 20 logical threads with 32k L1 cache, 256k L2 cache, and 25MB L3 cache. All of our experiments can be executed within 12 hours on a system with similar computational capabilities.

B Pseudocodes of the Considered Schemes

Algorithm 1: Score Climbing with the Single State Estimator

Input: MCMC kernel $K_{\boldsymbol{\lambda}}(\mathbf{z}, \cdot)$, initial sample $\mathbf{z}_0$, initial parameter $\boldsymbol{\lambda}_0$, number of iterations T, stepsize schedule γ_t

for $t = 1, 2, \ldots, T$ **do**
- $\mathbf{z}_t \sim K_{\lambda_{t-1}}(\mathbf{z}_{t-1}, \cdot)$
- $s(\boldsymbol{\lambda}; \mathbf{z}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$
- $g_{\text{single}} = s(\boldsymbol{\lambda}_{t-1}; \mathbf{z}_t)$
- $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g_{\text{single}}$

end

Algorithm 2: Score Climbing with the Sequential State Estimator

Input: MCMC kernel $K_{\boldsymbol{\lambda}}(\mathbf{z}, \cdot)$, initial sample $\mathbf{z}_0$, initial parameter $\boldsymbol{\lambda}_0$, number of iterations T, stepsize schedule γ_t

for $t = 1, 2, \ldots, T$ **do**
- $\tau = N(t-1)$
- **for** $i = 1, 2, \ldots, N$ **do**
  - $\mathbf{z}_t \sim K_{\lambda_{t-1}}(\mathbf{z}_{\tau+i}, \cdot)$
- **end**
- $s(\boldsymbol{\lambda}; \mathbf{z}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$
- $g_{\text{seq.}} = \frac{1}{N} \sum_{i=1}^{N} s(\boldsymbol{\lambda}_{t-1}; \mathbf{z}_{T+i})$
- $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g_{\text{seq.}}$

end

Algorithm 3: Score Climbing with the Parallel State Estimator

Input: MCMC kernel $K_{\boldsymbol{\lambda}}(\mathbf{z}, \cdot)$, initial samples $\mathbf{z}_0^{(1)}, \ldots, \mathbf{z}_0^{(N)}$, initial parameter $\boldsymbol{\lambda}_0$, number of iterations T, stepsize schedule γ_t

for $t = 1, 2, \ldots, T$ **do**
- **for** $i = 1, 2, \ldots, N$ **do**
  - $\mathbf{z}_t^{(i)} \sim K_{\lambda_{t-1}}(\mathbf{z}_{t-1}^{(i)}, \cdot)$
- **end**
- $s(\boldsymbol{\lambda}; \mathbf{z}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$
- $g_{\text{par.}} = \frac{1}{N} \sum_{i=1}^{N} s(\boldsymbol{\lambda}_{t-1}; \mathbf{z}_t^{(i)})$
- $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g_{\text{par.}}$

end

Algorithm 4: Conditional Importance Sampling Kernel

Input: previous sample $\mathbf{z}_{t-1}$, previous parameter $\boldsymbol{\lambda}_{t-1}$, number of proposals N

$\mathbf{z}^{(0)} = \mathbf{z}_{t-1}$

$\mathbf{z}^{(i)} \sim q_{\lambda_{t-1}}(\mathbf{z}) \quad \text{for } i = 1, 2, \ldots, N$

$w(\mathbf{z}^{(i)}) = p(\mathbf{z}^{(i)}, \mathbf{x}) \,/\, q_{\lambda_{t-1}}(\mathbf{z}^{(i)}) \quad \text{for } i = 0, 1, \ldots, N$

$\widetilde{w}^{(i)} = w(\mathbf{z}^{(i)}) / \sum_{i=0}^{N} w(\mathbf{z}^{(i)}) \quad \text{for } i = 0, 1, \ldots, N$

$\mathbf{z}_t \sim \text{Multinomial}(\widetilde{w}^{(0)}, \widetilde{w}^{(1)}, \ldots, \widetilde{w}^{(N)})$

Algorithm 5: Independent Metropolis-Hastings Kernel

Input: previous sample $\mathbf{z}_{t-1}$, previous parameter $\boldsymbol{\lambda}_{t-1}$,

$\mathbf{z}^* \sim q_{\lambda_{t-1}}(\mathbf{z})$

$w(\mathbf{z}) = p(\mathbf{z}, \mathbf{x}) / q_{\lambda_{t-1}}(\mathbf{z})$

$\alpha = \min\left(w\left(\mathbf{z}^*\right) / w\left(\mathbf{z}_{t-1}\right), 1\right)$

$u \sim \text{Uniform}(0, 1)$

if $u < \alpha$ **then**

  $\mathbf{z}_t = \mathbf{z}^*$

else

  $\mathbf{z}_t = \mathbf{z}_{t-1}$

end

C Probabilistic Models Considered in Section 4

C.1 Hierarchical Logistic Regression

The hierarchical logistic regression used in Section 4.3 is

$$
\begin{aligned}
\sigma_\beta &\sim \mathcal{N}^+(0, 1.0) \\
\sigma_\alpha &\sim \mathcal{N}^+(0, 1.0) \\
\boldsymbol{\beta} &\sim \mathcal{N}\left(\mathbf{0}, \sigma_\beta^2 \mathbf{I}\right) \\
\alpha &\sim \mathcal{N}\left(0, \sigma_\alpha^2\right) \\
p &\sim \mathcal{N}\left(\mathbf{x}_i^\top \boldsymbol{\beta} + \alpha,\ \sigma_\alpha^2\right) \\
y_i &\sim \text{Bernoulli-Logit}\left(p\right)
\end{aligned}
$$

where $\mathbf{x}_i$ and y_i are the predictors and binary target variable of the ith datapoints.

C.2 Gaussian Process Logistic Regression

The latent Gaussian process model used in Section 4.4 is

$$
\begin{aligned}
\log \alpha &\sim \mathcal{N}(0, 1) \\
\log \sigma &\sim \mathcal{N}(0, 1) \\
\log \ell_i &\sim \mathcal{N}(0, 1) \\
f &\sim \mathcal{GP}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\alpha^2, \sigma^2, \ell} + \delta\, \mathbf{I}\right) \\
y_i &\sim \text{Bernoulli-Logit}\left(f\left(\mathbf{x}_i\right)\right).
\end{aligned}
$$

The covariance $\boldsymbol{\Sigma}$ is computed using a kernel $k(\cdot, \cdot)$ such that $[\boldsymbol{\Sigma}]_{i,j} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$ where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points in the dataset. For the kernel, we use the Matern 5/2 kernel with automatic relevance determination (Neal, 1996) defined

as

$$k\left(\mathbf{x}, \mathbf{x}';\ \alpha^2, \sigma^2, \ell\right) = \alpha\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp\left(-\sqrt{5}r\right) \tag{16}$$

$$\text{where}\ \ r = \sum_{i=1}^{D} \frac{\left(\mathbf{x}_i - \mathbf{x}_i'\right)^2}{\ell_i^2} \tag{17}$$

where D is the number of dimensions. The jitter term δ is used for numerical stability. We set a small value of $\delta = 1 \times 10^{-6}$.

C.3 Radon Hierarchical Regression

The partially pooled linear regression model used in Section 4.5 is

$$\begin{aligned}
\sigma_{a_1} &\sim \text{Gamma}\left(\alpha = 1, \beta = 0.02\right) \\
\sigma_{a_2} &\sim \text{Gamma}\left(\alpha = 1, \beta = 0.02\right) \\
\sigma_y &\sim \text{Gamma}\left(\alpha = 1, \beta = 0.02\right) \\
\mu_{a_1} &\sim \mathcal{N}\left(0, 1\right) \\
\mu_{a_2} &\sim \mathcal{N}\left(0, 1\right) \\
a_{1,c} &\sim \mathcal{N}\left(\mu_{a_1}, \sigma_{a_1}^2\right) \\
a_{2,c} &\sim \mathcal{N}\left(\mu_{a_2}, \sigma_{a_2}^2\right) \\
y_i &\sim \mathcal{N}\left(a_{1,c_i} + a_{2,c_i}\, x_i,\ \sigma_y^2\right)
\end{aligned}$$

where $a_{1,c}$ is the intercept at the county c, $a_{2,c}$ is the slope at the county c, c_i is the county of the ith datapoint, x_i and y_i are the floor predictor of the measurement and the measured radon level of the $i$the datapoint, respectively. The model pools the datapoints into their respective counties, which complicates the posterior geometry (Betancourt, 2020).

D Proofs

Detailed derivation of ***Equation*** **(15)**

First, remember that the estimator is defined as

$$g_{\text{seq.}} = \frac{1}{N}\sum_{i=1}^{N} s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), \tag{18}$$

where $\mathbf{z}_{T+i} \sim K^i_{\boldsymbol{\lambda}_{t-1}}\left(\mathbf{z}_T, \cdot\right)$ and $\mathbf{z}_T$ is the last Markov-chain state at the previous SGD iteration $t-1$. Then, the variance is given as

$$\mathbb{V}\left[\, g_{\text{seq.}} \,\right] = \mathbb{V}\left[\mathbb{E}_{K(\mathbf{z}_T,\mathbf{z})}\left[\frac{1}{N}\sum_{i=1}^{N} s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \,\middle|\, \mathbf{z}_T \right] \right] + \mathbb{E}\left[\mathbb{V}_{K(\mathbf{z}_T,\mathbf{z})}\left[\frac{1}{N}\sum_{i=1}^{N} s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \,\middle|\, \mathbf{z}_T \right] \right] \qquad \text{(Total Variance)} \tag{19}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}_{K(\mathbf{z}_T,\mathbf{z})}\left[\, \mathbb{E}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] \right] \tag{20}$$

$$+ \mathbb{E}\left[\frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}_{K(\mathbf{z}_T,\mathbf{z})}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] + \frac{2}{N^2}\sum_{i<j}^{N} \operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right] \tag{21}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}_{K(\mathbf{z}_T,\mathbf{z})}\left[\, \mathbb{E}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] \right] \tag{22}$$

$$+ \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{E}_{K(\mathbf{z}_T,\mathbf{z})}\left[\, \mathbb{V}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] \right] + \frac{2}{N^2}\sum_{i<j}^{N} \mathbb{E}\left[\operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right], \tag{23}$$

where by assuming stationarity such that $\mathbf{z}_T \sim p\left(\mathbf{z} \mid \mathbf{x}\right)$,

$$= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}_{p(\mathbf{z}_T|\mathbf{x})}\left[\, \mathbb{E}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] \right] \qquad + \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{z}_T|\mathbf{x})}\left[\, \mathbb{V}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right) \mid \mathbf{z}_T \,\right] \right] \tag{24}$$

$$+ \frac{2}{N^2}\sum_{i<j}^{N} \mathbb{E}\left[\operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right] \tag{25}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N} \mathbb{V}_{p(\mathbf{z}|\mathbf{x})}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}\right) \right] + \frac{2}{N^2}\sum_{i<j}^{N} \mathbb{E}\left[\operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right] \qquad \text{(Total Variance)} \tag{26}$$

$$= \frac{1}{N}\mathbb{V}_{p(\mathbf{z}|\mathbf{x})}\left[\, s\left(\boldsymbol{\lambda}; \mathbf{z}\right) \right] + \frac{2}{N^2}\sum_{i<j}^{N} \mathbb{E}\left[\operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right] \tag{27}$$

$$= \frac{\sigma^2}{N} + \frac{2}{N^2}\sum_{i<j}^{N} \mathbb{E}\left[\operatorname{Cov}\left(s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+i}\right), s\left(\boldsymbol{\lambda}; \mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \right) \right] \tag{28}$$

Theorem 1. *(Bias of seq.-IMH) Assuming* $w^* = \sup_{\mathbf{z}} {}^{p(\mathbf{z}|\mathbf{x})}/_{q_{\lambda_t}(\mathbf{z})} < \infty$ *and the score function is bounded such that* $|\, s(\mathbf{z}; \boldsymbol{\lambda}) \,| \leq \frac{L}{2}$*, the bias of the sequential state estimator with an IMH kernel averaging N states at iteration t is bounded as*

$$\text{Bias}\left[g_{\text{seq.},t} \right] \leq L\left(1 - \frac{N}{2\,w^*}\right) + \mathcal{O}\left(\left(\frac{1}{w^*}\right)^2\right)$$

Proof of ***Theorem*** *1.* We employ a similar proof strategy with the works of Jiang et al. (2021, Theorem 4).

Let us first denote the empirical distribution of the Markov-chain states at iteration t as

$$\eta_{\text{seq.},t}(\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N} K^i(\mathbf{z}_T, \mathbf{z}), \tag{29}$$

where $\mathbf{z}_T$ is the last state of the Markov-chain at the previous SGD iteration. Now,

$$\left\| \eta_{seq.,t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} = \left\| \frac{1}{N}\sum_{i=1}^{N} K^i(\mathbf{z}_T, \cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \tag{30}$$

$$\leq \frac{1}{N}\sum_{i=1}^{N} \left\| K^i(\mathbf{z}_T, \cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \qquad \text{(Triangle inequality)} \tag{31}$$

For an IMH kernel with $w^* < \infty$, the geometric ergodicity of the IMH kernel (Mengersen and Tweedie, 1996, Theorem 2.1) gives the bound

$$\left\| K^t(\mathbf{z}_0, \cdot) - p(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} \leq \left(1 - \frac{1}{w^*}\right)^t. \tag{32}$$

For the SGD step t, $\boldsymbol{\lambda}_t$ is fixed, temporarily enabling ergodicity to hold. Therefore,

$$\begin{aligned}
\left\| \eta_{\mathrm{seq.},t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} &\leq \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{1}{w^*}\right)^i && (33)\\
&= \frac{1}{N} \sum_{i=1}^{N} C^i && (34)\\
&= \frac{1}{N} \left(\frac{C\left(1 - C^N\right)}{1 - C} \right) && (35)\\
&= \frac{C}{N} \frac{\left(1 - C^N\right)}{1 - C} && (36)\\
&= \frac{w^* - 1}{N} \left(1 - \left(1 - \frac{1}{w^*}\right)^N\right). && (37)
\end{aligned}$$

While this bound itself is not very intuitive, by performing a Laurent series expansiona at $x = \infty$, we obtain a close approximation

$$\frac{w^* - 1}{N} \left(1 - \left(1 - \frac{1}{w^*}\right)^N\right) = 1 - \frac{N+1}{2w^*} + \mathcal{O}\left(\left(\frac{1}{w^*}\right)^2\right). \tag{38}$$

Finally, by the definition of the total-variation distance,

$$\begin{aligned}
\mathrm{bias}\left[g_{\mathrm{seq.,t}}\right] &\leq \sup_{h:\mathcal{Z}\to[-L/2,L/2]} \left| \mathbb{E}_{\eta_{\mathrm{seq.},t}(\cdot)}\left[h\right] - \mathbb{E}_{p(\cdot|\mathbf{x})}\left[h\right] \right| && (39)\\
&= L \left\| \eta_{\mathrm{seq.},t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} && (40)\\
&\leq L\left(1 - \frac{N}{2\,w^*}\right) + \mathcal{O}\left(\left(\frac{1}{w^*}\right)^2\right). && (41)
\end{aligned}$$

□

Theorem 2. *(Bias of par.-IMH) Assuming* $w^* = \sup_{\mathbf{z}} {}^{p(\mathbf{z}|\mathbf{x})}/_{q_{\lambda_\tau}(\mathbf{z})} < \infty$ *and that the score function is bounded as* $|\, s\left(\mathbf{z};\boldsymbol{\lambda}\right)| \leq \frac{L}{2}$*, the bias of the parallel state estimator with an IMH kernel and N parallel chains at iteration t is bounded as*

$$\mathrm{Bias}\left[g_{\mathrm{par.,t}}\right] \leq L\left(1 - \frac{1}{w^*}\right).$$

Proof of ***Theorem*** *2.* We denote the empirical distribution of the Markov-chain states at iteration t as

$$\eta_{\mathrm{par.},t}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} K\left(\mathbf{z}_{t-1}^{(i)}, \mathbf{z}\right). \tag{42}$$

Similarly with Theorem 1,

$$\begin{aligned}
\left\| \eta_{\mathrm{par.},t}(\mathbf{z}) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} &= \left\| \frac{1}{N} \sum_{i=1}^{N} K\left(\mathbf{z}_{t-1}^{(i)}, \mathbf{z}\right) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} && (43)\\
&\leq \frac{1}{N} \sum_{i=1}^{N} \left\| K\left(\mathbf{z}_{t-1}^{(i)}, \cdot\right) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} && \text{(Triangle inequality)} \quad (44)\\
&= \left\| K(\mathbf{z}_{t-1}, \cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\mathrm{TV}} && \text{(Uniform ergodicity)} \quad (45)\\
&\leq 1 - \frac{1}{w^*}. && (46)
\end{aligned}$$

And, finally the bias is given as

$$\text{Bias}\left[g_{\text{par.,t}}\right] \leq \sup_{h:\mathcal{Z}\to[-L/2,L/2]} \left| \mathbb{E}_{\eta_{\text{par.},t}(\cdot)}\left[h\right] - \mathbb{E}_{p(\cdot|\mathbf{x})}\left[h\right] \right| \tag{47}$$

$$= L \left\| \eta_{\text{par.},t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \tag{48}$$

$$\leq L\left(1 - \frac{1}{w^*}\right). \tag{49}$$

□

Theorem 3. *(Bias of single-CIS) For a CIS kernel with N internal proposals, assuming* $w^* = \sup_{\mathbf{z}} {}^{p(\mathbf{z}|\mathbf{x})}/_{q_\lambda(\mathbf{z})} < \infty$ *for* $\forall\boldsymbol{\lambda}$*,* $N > 2$*, and that the score function is bounded such that* $|\, s\left(\mathbf{z};\boldsymbol{\lambda}\right)| \leq \frac{L}{2}$*, the bias of the single state estimator at iteration t is bounded as*

$$\text{Bias}\left[g_{\text{cis.,t}}\right] \leq L\left(1 - \frac{N-1}{2\,w^* + N - 2}\right)$$

Proof of ***Theorem*** *3.* Let us first denote the empirical distribution of the Markov-chain states at iteration t as

$$\eta_{\text{cis.},t}(\mathbf{z}) = K\left(\mathbf{z}_{t-1}, \mathbf{z}\right), \tag{50}$$

and consequently,

$$g_{\text{cis},t}(\boldsymbol{\lambda}) = \int s\left(\mathbf{z};\boldsymbol{\lambda}\right)\, \eta_{\text{cis.},t}(\mathbf{z})\, d\mathbf{z}. \tag{51}$$

The CIS sampler is identical to the iterated sampling importance resampling (i-SIR) algorithm described by Andrieu et al. (2018). They showed that the i-SIR kernel achieves a geometric convergence rate such that

$$\left\| K^t\left(\mathbf{z}_{t-1}, \cdot\right) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \leq \left(1 - \frac{N-1}{2\,w^* + N - 2}\right)^t. \tag{52}$$

From this, the bound can be shown as

$$\text{bias}\left[g_{\text{cis.,t}}\right] \leq \sup_{h:\mathcal{Z}\to[-L/2,L/2]} \left| \mathbb{E}_{\eta_{\text{cis.},t}(\cdot)}\left[h\right] - \mathbb{E}_{p(\cdot|\mathbf{x})}\left[h\right] \right| \tag{53}$$

$$= L \left\| \eta_{\text{cis.},t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \tag{54}$$

$$\leq L\left(1 - \frac{N-1}{2\,w^* + N - 2}\right) \tag{55}$$

given that $N > 2$. □

Proposition 1. $w^* = \sup_{\mathbf{z}} {}^{p(\mathbf{z}|\mathbf{x})}/_{q_\lambda(\mathbf{z})}$ *is bounded below exponentially by the KL divergence such that*

$$\exp\left(D_{\text{KL}}(p\left(\cdot \mid \mathbf{x}\right) \parallel q_\lambda\left(\cdot\right))\right) \leq w^*.$$

Proof of ***Proposition*** *1.*

$$D_{\text{KL}}(p\left(\cdot \mid \mathbf{x}\right) \parallel q_\lambda\left(\cdot\right)) \tag{56}$$

$$= \int p\left(\mathbf{z} \mid \mathbf{x}\right) \log \frac{p\left(\mathbf{z} \mid \mathbf{x}\right)}{q_\lambda\left(\mathbf{z}\right)}\, d\mathbf{z} \tag{57}$$

$$\leq \int p\left(\mathbf{z} \mid \mathbf{x}\right) \log w^*\, d\mathbf{z} \tag{58}$$

$$= \log w^* \tag{59}$$

□