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 a critical part of the problem. This fact had lead to a quest for good divergence measures (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 performing inclusive VI, 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 gradient estimated using Markov-chain Monte Carlo (MCMC). These MCMC score climbing schemes operate a (or multiple) Markov-chain in conjuction with the VI optimizer. In addition, within the MCMC kernel, they both use Metropolis-Hastings proposals generated from the variational approximation $\mathbf{z}^* \sim q_{\lambda}(\cdot)$. Therefore, the MCMC kernel itself benefits from the variational inference process, gradually improving over time. This enables the MCMC kernel to be sufficiently efficient without the need of computationally expensive kernel such as Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2011b; Betancourt, 2017).

While the methods by Naesseth et al. and Ou and Song are conceptually similar, they use their MCMC kernels in slightly different ways. At each SGD iteration, for estimating the score function, Naesseth et al. use a single state of the Markov-chain generated from a computationally expensive MCMC kernel. On the other hand, given a similar computational budget, Ou and Song use N states generated from a cheaper MCMC kernel. We call the

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

former option the *singlestateestimator* and the later the *sequentialstateestimator* It is now natural to ask, “which is better? An estimator with multiple cheap samples? or one with a single expensive sample?”.

Furthermore, a third, but seldomly mentioned option exist. We call this the setup *the parallel state estimator*. For this estimator, we operate N parallel Markov-chains parallel, where only a single state transition is performed on each chain. The variance of this estimator dramatically benefits from increasing the computational budget compared to the single and sequential state estimators. While this comes at the cost of slightly higher bias, we observe that the variance reduction is significant enough.

In this work, we compare the bias and variance of the three different schemes of using an MCMC kernel for score climbing variational inference. We discuss the bias and variance of the three schemes, and conduct experiments on general Bayesian inference benchmarks. Our results show that, given a similar number of computational budget N, the parallel state estimator results in the best performance. Also compared to exclusive KL divergence minimization methods such as automatic differentiation VI, score climbing VI is much more efficient since we do not need to differentiate through the likelihood.

Contribution Summary

- We propose the parallel state estimator for estimating the score function using MCMC (**??**).
- 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 benchmark problems (**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), which requires unbiased gradient estimates of the optimization target $g(\boldsymbol{\lambda})$. In this case, SGD can be performed by repeating the update

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

where $\gamma_1, \ldots, \gamma_T$ is a step-size schedule follwing the conditions of Robbins and Monro (1951); Bottou (1999). In the case of inclusive variational inference, 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

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

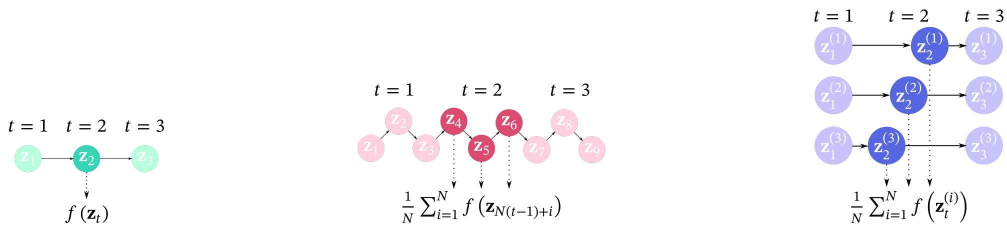
where $w(\mathbf{z}) = p(\mathbf{z}, \mathbf{x}) / q_{\boldsymbol{\lambda}}(\mathbf{z})$ is known as the *importance weight*, and $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$ are N independent samples from $q_{\boldsymbol{\lambda}}(\mathbf{z})$. This scheme is equivalent to adaptive IS methods (Cappé et al., 2008; Bugallo et al., 2017) since the IS proposal $q_{\boldsymbol{\lambda}}(\mathbf{z})$ is iteratively optimized based on the current samples. Though 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 is known as 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 Variational Inference

3.1 Overview of Estimation Strategies

Overview Recently, Naesseth et al. (2020) and Ou and Song (2020) proposed two similar but independent methods for performing inclusive variational inference. Both methods estimate the score gradient by operating a Markov-chain in parallel with the VI optimization sequence. Also, they both use MCMC kernels that can effectively used the variational approximation $q_{\boldsymbol{\lambda}_t}(\mathbf{z})$. Because of this, both methods are much more efficient compared to 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. estimate the score gradient by performing an MCMC iteration and update the parameters such that



(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 denote the MCMC samples used for estimating the score gradient at $t = 2$.

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

where $K_{\lambda_{t-1}}(\mathbf{z}_{t-1}, \cdot)$ is a MCMC kernel leaving $p(\mathbf{z} \mid \mathbf{x})$ invariant and $g_{\text{single}}(\boldsymbol{\lambda})$ denotes the score estimator. For $K_{\lambda_{t-1}}(\mathbf{z}_{t-1}, \cdot)$, 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 $q_{\lambda_{t-1}}(\mathbf{z})$, hence the dependence on $\boldsymbol{\lambda}_{t-1}$. When compared to MCMC kernels that only use a single sample from $q_{\lambda_{t-1}}(\mathbf{z})$, 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 $i \in \{1, \ldots, N\}$,

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

where $\mathbf{z}_T$ is the last Markov-chain state of the previous SGD iteration. $K^{i}_{\mathbf{z}_{t-1}}(\mathbf{z}_T, \cdot)$ denotes the MCMC kernel sequentially applied i times. For the MCMC kernel, they use the classic independent Metropolis-Hastings (IMH, Robert and Casella 2004, Algorithm 25 Hastings 1970) algorithm, which uses only a single sample from $q_{\lambda_{t-1}}(\mathbf{z})$ (notice the dependence on $\boldsymbol{\lambda}_{t-1}$ just like the aforementioned CIS kernel). Therefore, the cost of N state transitions with IMH is similar to the cost of a single transition with CIS. Since the estimator uses sequential states, we call it the sequential state esimator 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 in an expensive way, while the latter uses multiple samples generated in a cheap way. Illustrations of the two schemes are provided in Figures 1a and 1b.

Parallel State Estimator In this work, we add 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 $i \in \{1, \ldots, N\}$,

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

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-chain 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 Detailed pseudocodes of the considered schemes are provided in the *supplementary material*.

Computational Cost The three scheme using the CIS kernel and the IMH kernel can have different computational cost depending on the parameter N. The computational costs of each scemes are organized in Table 1. In the CIS kernel, N controls the number of internal proposals sampled from $q_{\lambda}(\mathbf{z})$. In the sequential and parallel state estimators, the IMH kernel only uses a single sample from $q_{\lambda}(\mathbf{z})$, but applies the kernel N times. When estimating the score, the single state estimator computes $\nabla_{\lambda} \log q_{\lambda}(\mathbf{z})$ 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.

Table 1: Computational Cost of Markov-chain Schemes

	Estimation			Stochastic gradient	
	$p(\mathbf{z}, \mathbf{x})$ # Eval.	$q_{\lambda}(\mathbf{z})$ # Eval.	$q_{\lambda}(\mathbf{z})$ # Samples	$p(\mathbf{z}, \mathbf{x})$ # Grad.	$q_{\lambda}(\mathbf{z})$ # Grad.
ADVI	0	0	N	N	0
Single state estimator with CIS	$N-1$	N	$N-1$	0	1^1 or N^2
Sequential state estimator with IMH	N	N	N	0	N
Parallel state estimator with IMH	N	N	N	0	N

* N is the number of samples used in each method.

[1] Vanilla CIS kernel.

[2] Rao-Blackwellized CIS kernel.

3.2 Theoretical Analysis of Bias

Adaptive MCMC and Ergodicity For bounded functions, a bound on the bias of MCMC estimators can be easily derived from the convergence rates of MCMC kernels as shown by Jiang et al. (2021, Theorem 4). In the context of MSC, the convergence rate of an MCMC kernel is a subtle subject since the kernel is now *adaptive* as it depends on $\boldsymbol{\lambda}_t$, which is in turn dependent 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). However, our setting crucially differs with adaptive MCMC in that our goal is not to obtain asymptotically unbiased samples. Instead, we use the MCMC samples acquired during each SGD step, in which $\boldsymbol{\lambda}_t$ is fixed. That is, our MCMC kernel is instantaneously not adaptive, and we are thus we are free to use the ergodicity results of these kernels. However, we note that, as far as Deoblin's condition holds such that $w^* = \sup_{\mathbf{z},\boldsymbol{\lambda}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}}(\mathbf{z}) < \infty$ and the SGD stepsize sequence satisfies the diminishing adaptation condition (Roberts and Rosenthal, 2007), the MCMC kernel will indeed result in asymptotically unbised samples.

Boundedness Assumption Since convergence rates are defined with the total-variation distance, our bias results assume that the score function is bounded. That is, $\| \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z}) \| < L$ for any $\boldsymbol{\lambda}$. This boundedness assumption is reasonable since theoretical guarentees of SGD often assume Lipschitz-continuity of the gradients, from which boundedness follows as a consequence.

Assuming $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}_t}(\mathbf{z}) < \infty$ for $\forall \boldsymbol{\lambda}$ 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 at iteration t is bounded as

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

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

Theorem 1. *Assuming $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}_{\tau}}(\mathbf{z}) < \infty$ for $\forall \boldsymbol{\lambda}$ 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 at iteration t is bounded as*

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

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

Finally, we analyze the bias of 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). Especially, we utilize the the convergence rate of the i-SIR kernel. In addition, we note that the CIS kernel can be reformulated as an accept-reject type kernel that uses Barker's acceptance function (Barker, 1965). With this perspective, it is identical to the ensemble MCMC sampler independently proposed by Austad (2007); Neal (2011a). It can also be found in the review on multiple-try MCMC methods by Martino (2018, Table 12).

Theorem 2. *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*

$$\mathrm{Bias}\left[g_{\mathrm{cis.},t}\right] \leq L\,C \quad \textit{where} \quad C = \left(1 - \frac{N}{w^*}\right) < 1.$$

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

Reducing Bias by Increasing N Our results suggest that, for the seq.-IMH estimator and single-CIS estimator, increasing N improves the bias decrease rate. However, it is important to note that all bias bounds depend on w^*. By the following proposition, in the initial stages of VI where the KL divergence is large, w^* is bounded below exponentially.

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

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

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

Thus, in the initial steps of VI, the bias decrease rate C will be close to 1, making the effect of N minimal. On the

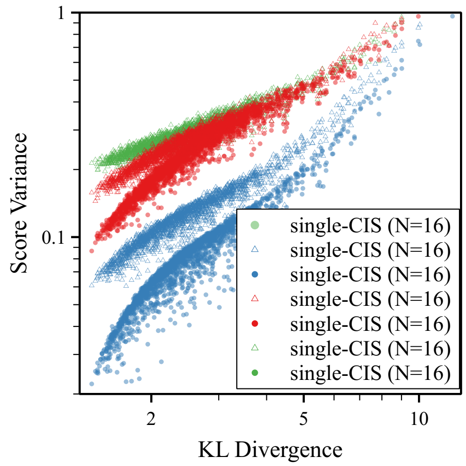


Figure 2: Variance of score function estimated using the three estimators depending on N and the KL divergence.

other hand, in the later steps of VI, the KL divergence is small. However, in this case t will be larege, therefore making the bias small regardless.

3.3 Theoretical Analysis of Variance

Geometric ergodicity of the CIS and IMH kernels guarentee that the bias will be small regardless of the kernel and parameter N. In contrast, variance often dominate the mean-square error of MCMC estimators. Therefore, analyzing the variance will be more relevant in practice.

Theorem 3. *The variance of the sequential state estimator is*

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

Theorem 4. *The variance of the single mode estimator with a CIS kernel* $\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right]$ *is approximately bounded below such that*

$$\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right] \geq \frac{N^4 Z^4}{(w(\mathbf{z}_{t-1}) + NZ)^4} \mathbb{V}_{q_\lambda}\left[\, f_{IS} \,|\, \mathbf{z}_{t-1} \,\right], \quad (10)$$

where $Z = \mathbb{E}_{q_\lambda}[p(\mathbf{z}, \mathbf{x})/q_\lambda(\mathbf{z})] = \int p(\mathbf{z}, \mathbf{x}) d\mathbf{z}$ *is the normalizing constant.*

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

4 Evaluations

4.1 Numerical Simulation

Experimental Setup We first present numerical simulation results of the three estimators. We chose the target distribution $p(\mathbf{z})$ to be a 10 dimensional white Gaussian. We then randomly generated 2048 random $q_{\mu,\Sigma}(\mathbf{z})$ where $\boldsymbol{\mu}$ 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 compute the bias and variance of the score function.

Results The variance results are shown in Figure 2. We do not present the bias as the three schemes were visually indistinguishable for all settings. From the results, when the KL divergence is large, we can see that seq.-IMH and single-CIS do not benefit from increasing N. On the other hand, the parallel state estimator always benefit from increasing N.

4.2 Baselines and Implementation

Implementation We implemented MSC with PIMH 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. We set the computational budget $N = 10$ and $T = 10^4$ for all experiments unless specified.

Considered Baselines We compare

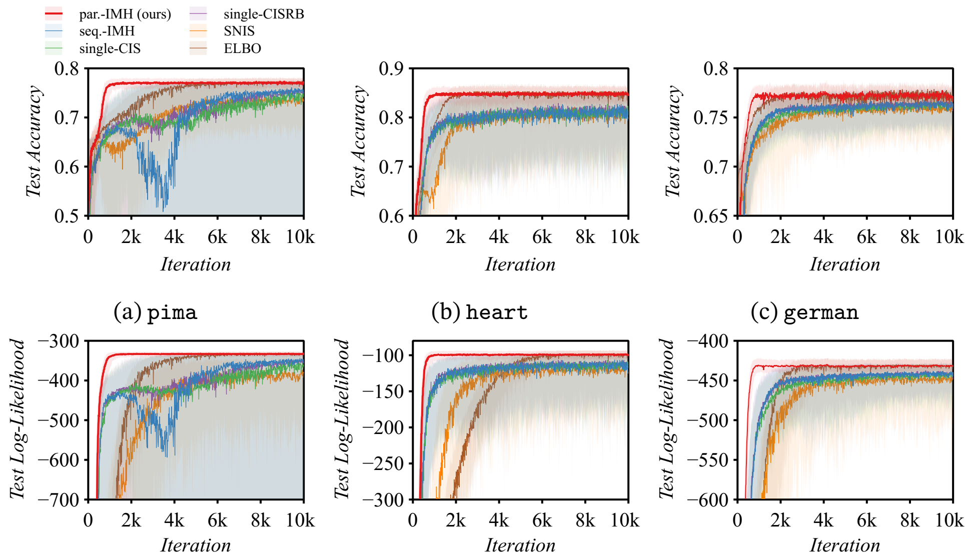
❶ score climbing with the parallel state estimator and the IMH kernel (**par.-IMH**),
❷ score climbing with the sequential state estimator and the IMH kernel (**seq.-IMH**),
❸ score climbing with the single state estimator and the CIS kernel (**single-CIS**, Naesseth et al. 2020),
❹ score climbing with the single state estimator, the CIS kernel, and Rao-Blackwellization (**MSC-CISRB**, Naesseth et al. 2020),
❺ the adaptive IS method using SNIS as introduced in Section 2.1 (**SNIS**),
❻ evidence lower-bound maximization (**ELBO**, Ranganath et al. 2014).

Specifically, we use automatic differentiation VI (ADVI, Kucukelbir et al. 2017) implemented by Turing.

4.3 Hierarchical Logistic Regression

Experimental Setup We evaluate MSC-PIMH on logistic regression with the Pima Indians diabetes (`pima`, $\mathbf{z} \in \mathbb{R}^{11}$, Smith et al. 1988), German credit (`german`, $\mathbf{z} \in \mathbb{R}^{27}$), and heart disease (`heart`, $\mathbf{z} \in \mathbb{R}^{16}$, 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 used in VI, we choose



(a) `pima` (b) `heart` (c) `german`

(d) `pima` (e) `heart` (f) `german`

Figure 3: Test accuracy and log-likelihood of logistic regression problems. The solid lines and colored regions are the medians and 80% percentiles computed from 100 repetitions.

a more complex hierarchical logistic regression model

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

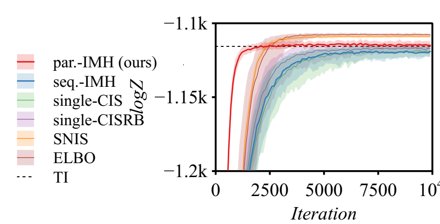
where $\mathcal{N}^+(\mu, \sigma)$ is a positive constrained normal distribution with mean μ and standard deviation σ, $\mathbf{x}_i$ and y_i are the feature vector and target variable of the ith datapoint. The extra degrees of freedom σ_β and σ_α make this model relatively more challenging.

Results The test accuracy and test log-likelihood results are shown in Figure 4. Our proposed MSC-PIMH is the fastest to converge on all the datasets. Despite having access to high-quality HMC samples, RWS fails to achieve a similar level of performance to MSC-PIMH. However, RWS converges faster than MSC-CIS and MSC-CISRB. Among the two, MSC-CISRB performs only marginally better than MSC-CIS. Meanwhile, SNIS converges the most slowly among inclusive VI methods. Although much slower to converge, ELBO achieves competitive results.

Inclusive VI v.s. Exclusive VI The results of Figure 4 might be misleading to conclude that inclusive and exclusive VI deliver similar results. However, in the parameter space, they choose different optimization paths. This is shown in **??** through the Pareto-$\hat{k}$ diagnostic (Dhaka et al., 2020; Vehtari et al., 2021), which determines how reliable the importance weights are when computed using $q_\lambda(\mathbf{z})$. While the test accuracy suggests that ELBO converges around $t = 2000$, in terms of Pareto-$\hat{k}$, it takes much longer to converge (about $t = 5000$). This shows that, even if their predictive performance is similar, the inclusive VI chooses paths that have better density coverage as expected.

4.4 Gaussian Process Classification

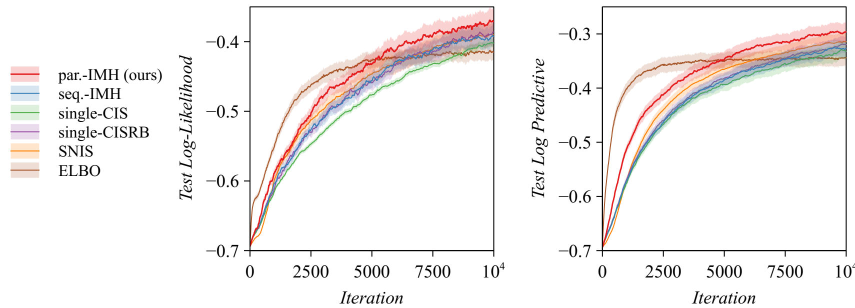
4.5 Marginal Likelihood Estimation



(a) `radon`

Figure 5: Marginal log-likelihood ($\log Z$) estimates of considered methods. ELBO is omitted in **??** as it failed to deliver reasonable estimates. The solid lines and colored regions are the medians and 80% percentiles computed from 100 repetitions.

Experimental Setup 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,



(a) `sonar` (b) `ionosphere`

Figure 4: Test accuracy and log-likelihood of logistic regression problems. The solid lines and colored regions are the medians and 80% percentiles computed from 100 repetitions.

2017).

Results The results are shown in Figure 5. On `radon`, MSC-PIMH converges quickly and provides the most accurate estimate. By contrast, MSC-CIS and MSC-CISRB 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 is a method for minizing the inclusive KL divergence. Concurrently, Ou and Song (2020) has proposed JSA which is conceptually very similar to MSC. Unlike Naesseth et al., they apply JSA for training variational autoencoders with discrete latent variables. Also, JSA can only be applied to models with *i.i.d.*, which is more restrictive. Other than MSC and JSA only a few have been proposed for general VI based on 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 Other than for minimizing the inclusive KL, MCMC has been widely utilized for VI. For example, Ruiz and Titsias (2019); Salimans et al. (2015) construct alternative divergence bounds from samples from an MCMC sampler. For the MCMC kernel, both caess used the costly HMC kernel. It would be interesting to investigate whether these method would benefit from different types of MCMC kernels and estimation schemes as discussed in this work.

Adaptive MCMC As pointed out by Ou and Song (2020), MSC is structurally equivalent to adaptive MCMC methods. Strong resemblence can be found in methods using stochastic approximation for adapting the proposal distribution used inside the MCMC kernel. In particular, Andrieu and Thoms (2008); Garthwaite et al. (2016) discuss the use of stochastic approximation in adaptive MCMC. Among adaptive MCMC methods, Andrieu and Moulines (2006); Keith et al. (2008); Holden et al. (2009); Giordani and Kohn (2010) specifically discuss adapting the propsosal of IMH kernels. In particular, Keith et al. (2008) propose to use *cross-entroy 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, **??** have proposed to minimize various other divergence measures.

6 Conclusions

In this paper, we investigated the properties of Markovian score climbing (MSC) with independent Metropolis-Hastings (IMH) type Markov-chain Monte Carlo (MCMC) kernels. We proved that IMH type kernels are able to automatically perform bias-variance tradeoff using their accept-reject mechanism. We also analyzed the limitation of the conditional importance sampling (CIS) kernel originally used in MSC. We then proposed parallel IMH (PIMH) as an alternative that enjoys the benefits of CIS without its limitations. Our experiments verify that MSC combined with PIMH performs well on the considered Bayesian inference problems, even compared to exclusive variational inference methods.

Acknowledgments

This work initially started in the process of understanding the performance of the Markovian score climbing algorithm by Naesseth et al. (2020). We thank Hongseok Yang for pointing us to a relevant related work by Kim et al. (2021), Guanyang Wang for insightful discussions about the independent Metropolis-Hastings algorithm, Geon Park and Kwanghee Choi for constructive comments that enriched this paper. We also aknowledge the Computer Science Department of Sogang University for providing computational resources.

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.

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., Andersen, M. R., Magnusson, M., Huggins, J., and Vehtari, A. (2020). Robust, accurate stochastic optimization for variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 10961–10973. Curran Associates, Inc.

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.

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.

Minh, D. D. L. and Minh, D. L. P. (2015). Understanding the Hastings Algorithm. *Communications in Statistics - Simulation and Computation*, 44(2):332–349.

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. (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]*.

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*.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612.

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.

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.

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.

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.

Stan Development Team (2020). Stan modeling language users guide and reference manual, version 2.23.0.

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2021). Pareto smoothed importance sampling. *arXiv:1507.02646 [stat]*.

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.

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.

A Relationship Between Sampling Methods

We organize the sampling methods described in this work in Table 2.

Table 2: Comparison of Sampling Method Designs

Algorithm	Origin	Proposal	M-H Test	Acceptance Ratio	Multiple Proposals	Reference
RWMH[1]	MCMC	Dependent	✓	M-H	✗	
HMC	MCMC	Dependent	✓	M-H	✗	Duane et al. (1987)
SNIS	IS	Independent	✗		✓	
IMH	MCMC	Independent	✓	M-H	✗	
CIS	PMCMC[4]	Independent	✓	Barker	✓	Naesseth et al. 2020
En. MCMC[2]	MCMC	Both	✓	Barker	✓	Neal 2011a
PMP MCMC[3]	MCMC	Dependent	✓	Barker	✓	Austad 2007

[1] Random-walk Metropolis-Hastings
[2] Ensemble MCMC
[3] Parallel multiple proposals MCMC
[4] Particle MCMC

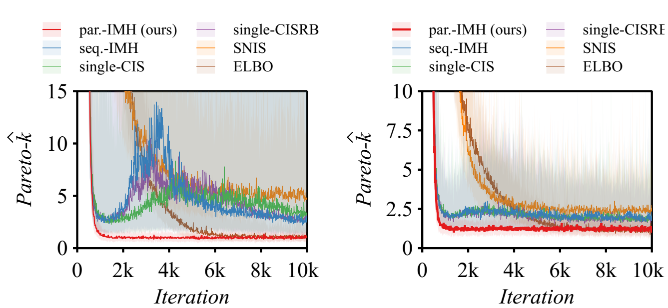
In this paper, we designated kernels that use independent proposals and perform a Metropolis-Hastings (M-H) test as "IMH type" kernels. While the original paper of CIS does not mention it as an IMH type, we have shown in **??** that it is indeed an IMH type kernel that uses Barker's acceptance ratio and multiple proposals per transition. This, in turn, reveals close connections with ensemble MCMC by Neal (2011a). While parallel multiple proposals MCMC by Austad (2007) also uses Barker's acceptance ratio and multiple proposals, it only considers dependent proposals, unlike ensemble MCMC. Although in principle, it should work with independent proposals without modification.

B Additional Experimental Results

B.1 Experimental Environment

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 a few days on a system with similar computational capabilities.

B.2 Additional Results of Logistic Regression Experiments



(a) Pima (b) Heart

Figure 6: Pareto-$\hat{k}$ results of logistic regression problems. The solid lines are the median of 100 repetitions while the colored regions are the 80% empirical percentiles.

C Pseudocodes of the Considered Schemes

Algorithm 1: Single State Estimator

Input: MCMC kernel $K(\mathbf{z}, \cdot)$, 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**
- $\mathbf{z}_t \sim K(\mathbf{z}_{t-1}, \cdot)$
- $s(\mathbf{z}; \boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$
- $g_{\text{single}} = s(\mathbf{z}_t; \boldsymbol{\lambda}_{t-1})$
- $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g_{\text{single}}$

end

Algorithm 2: Sequential State Estimator

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

end

Algorithm 3: Parallel State Estimator

Input: 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**
  - $\mathbf{z}_t^{(i)} \sim K(\mathbf{z}_{t-1}^{(i)}, \cdot)$
- **end**
- $s(\mathbf{z}; \boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} \log q_{\boldsymbol{\lambda}}(\mathbf{z})$
- $g_{\text{par.}} = \frac{1}{N} \sum_{i=1}^{N} s(\mathbf{z}_t^{(i)}; \boldsymbol{\lambda}_{t-1})$
- $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, g_{\text{par.}}$

end

Algorithm 4: Conditional Importance Sampling Kernel

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

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

$\mathbf{z}^{(i)} \sim q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z})$ for $i = 1, 2, \ldots, N$

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

$\widetilde{w}^{(i)} = w(\mathbf{z}^{(i)}) / \sum_{i=0}^{N} w(\mathbf{z}^{(i)})$ 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 $\boldsymbol{\lambda}_{t-1}$,

$\mathbf{z}^* \sim q_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z})$

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

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

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

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

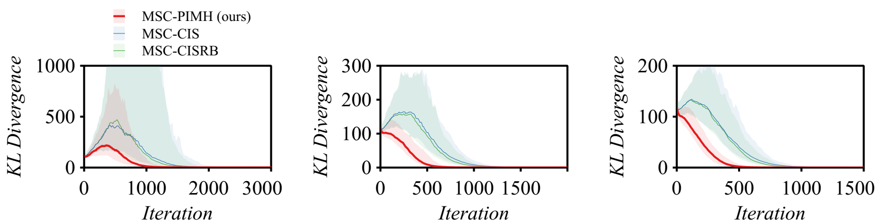
  $\mathbf{z}_t = \mathbf{z}^*$

else

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

end

C.1 Isotropic Gaussian Experiments



(a) $N = 2^2$ (b) $N = 2^4$ (c) $N = 2^6$

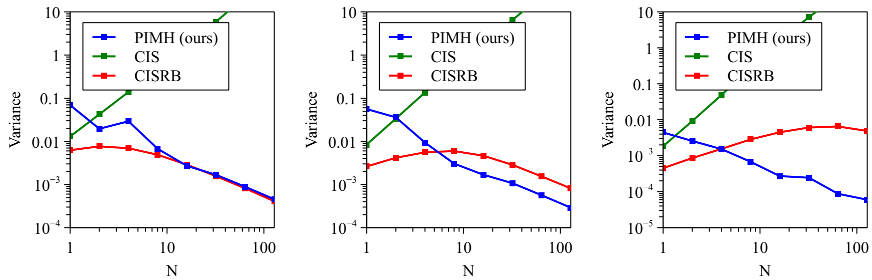
Figure 7: 100-D isotropic Gaussian example with a varying computational budget N. MSC-PIMH converges faster than MSC-CIS and MSC-CISRB regardless of N. Also, the convergence of MSC-PIMH becomes more stable/monotonic as N increases. The solid lines and colored regions are the medians and 80% percentiles computed from 100 repetitions.

We perform experiments with a 100-D isotropic multivariate Gaussian distribution. With Gaussian distributions, convergence can be evaluated exactly since their KL divergence is available in a closed form. We compare the performance of MSC-PIMH, MSC-CIS, and MSC-CISRB with respect to the N (number of proposals for MSC-CIS, MSC-CISRB; number of parallel chains for MSC-PIMH). The results are shown in Figure 7. While MSC-PIMH shows some level of overshoot wih $N = 4$, it shows monotonic convergence with larger N. On the other hand, both MSC-CIS and MSC-CISRB overshoots even with $N = 64$. This clearly shows that PIMH enjoys better gradient estimates compared to the CIS kernel.

D Numerical Simulation

We present numerical simulations of our analyses in **??** and **??**. In particular, we visualize the fact that the variance of the CIS kernel can increase with the number of proposals N when the KL divergence is large, as described in (**??**).

Experimental Setup We first set the target distribution as $p(z \mid x) = \mathcal{N}(0, 1)$ and the proposal distribution as $q(z; \mu) = \mathcal{N}(\mu, 2)$ with varying mean. We measure the variance of estimating the score function $s(z, \mu) = \frac{\partial q(\mathbf{z}; \mu)}{\partial \mu}$ using the CIS, CISRB, and PIMH kernels, given the previous Markov-chain denoted bystate z_{t-1} and computational budget N. For CIS and CISRB, we set a fixed z_{t-1}, while for PIMH, we randomly sample N samples from $\mathbf{z}_{t-1} \sim p(z \mid z)$ (we obtained similar trends regardless of the distribution of z_{t-1}). The variance is estimated using 2^{14} samples from $K(\mathbf{z}_{t-1}, \cdot)$. We report the variance across varying N and varying KL divergence between $q_{\lambda}(z)$ and $p(\mathbf{z} \mid \mathbf{x})$. The latter is performed by varying the difference between the mean of the proposal and the target distributions denoted by $\Delta\mu = \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[\, z \,\right] - \mathbb{E}_{q_{\lambda}}\left[\, z \,\right]$.



(a) $\Delta\mu = 0$ (b) $\Delta\mu = 2$ (c) $\Delta\mu = 4$

Figure 8: Conditional variance of different MCMC kernels with varying N and varying difference between the mean of the target and proposal distributions.

Results Summary The results are presented in Figure 8. We can see that, when the difference of the mean of the p and q is large, the variance of CISRB *increases* with N. This increasing trend becomes stronger as the KL divergence between p and q increases. While this simulation suggests that CISRB has much smaller variance compared to CIS, our realistic experiments in Section 4 did not reveal such levels of performance gains. Is also visible that PIMH has a slightly larger variance compared to CIS in the small N regime. This is due to the higher acceptance rate of the Metropolis-Hastings acceptance ratio used by PIMH compared to Barker's acceptance ratio used by CIS (Peskun, 1973; Minh and Minh, 2015).

E Probabilistic Models Considered in Section 4

E.1 Hierarchical Logistic Regression

The hierarchical logistic regression used in Section 4.3 is

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

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

E.2 Stochastic Volatility

The stochastic volatility model used in Section 4.5 is

$$\begin{aligned}
\mu &\sim \text{Cauchy}(0, 10) \\
\phi &\sim \text{Uniform}\,(-1, 1) \\
\sigma &\sim \text{Cauchy}^+(0, 5) \\
h_1 &\sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \\
h_{t+1} &\sim \mathcal{N}\left(\mu + \phi\,(h_t - \mu), \sigma^2\right) \\
y_t &\sim \mathcal{N}\left(0, \exp\left(h_t\right)\right)
\end{aligned}$$

where y_t is the stock price at the tth point in time. We used the reparameterized version where h_t is sampled from a white multivariate Gaussian described by the Stan Development Team (2020).

E.3 Radon Hierarchical Regression

The partially pooled linear regression model used in Section 4.5 is

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

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).

F Proofs

Assuming $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\lambda_t}(\mathbf{z}) < \infty$ for $\forall \boldsymbol{\lambda}$ 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 at iteration t is bounded as

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

Proof of section 3.2. 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

$$\eta_{\text{seq.},t}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} K^i(\mathbf{z}_T, \mathbf{z}), \tag{11}$$

where $\mathbf{z}_T$ is the last state of the Markov-chain at the previous SGD iteration. Consequently, the estimator can be described as

$$g_{\text{seq},t}(\boldsymbol{\lambda}) = \int s\,(\mathbf{z}; \boldsymbol{\lambda})\; \eta_{\text{seq.},t}(\mathbf{z})\, d\mathbf{z}. \tag{12}$$

Now,

$$\begin{aligned}
\left\| \eta_{seq.,t}(\cdot) - p\,(\cdot \mid \mathbf{x}) \right\|_{\text{TV}} &= \left\| \frac{1}{N} \sum_{i=1}^{N} K^i(\mathbf{z}_T, \cdot) - p\,(\cdot \mid \mathbf{x}) \right\|_{\text{TV}} && (13) \\
&\leq \frac{1}{N} \sum_{i=1}^{N} \left\| K^i(\mathbf{z}_T, \cdot) - p\,(\cdot \mid \mathbf{x}) \right\|_{\text{TV}} \qquad \text{(Triangle inequality)} && (14)
\end{aligned}$$

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

$$\left\| K^t(\mathbf{z}_0, \cdot) - p(\cdot \mid \mathbf{x}) \right\|_{\text{TV}} \leq \left(1 - \frac{1}{w^*}\right)^t. \tag{15}$$

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

$$\left\| \eta_{\text{seq.},\,t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \leq \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{1}{w^*}\right)^i \tag{16}$$

$$= \frac{1}{N} \sum_{i=1}^{N} C^i \tag{17}$$

$$= \frac{1}{N} \left(\frac{C\left(1 - C^N\right)}{1 - C} \right) \tag{18}$$

$$= \frac{C}{N} \frac{\left(1 - C^N\right)}{1 - C} \tag{19}$$

$$\leq \frac{1}{N} \frac{C}{1 - C} \tag{20}$$

$$= \frac{1}{N} \frac{1 - 1/w^*}{1/w^*} \tag{21}$$

$$= \frac{1}{N} \left(w^* - 1\right) \tag{22}$$

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

$$\text{bias}\left[g_{\text{seq.,t}}\right] \leq \left\| \eta_{seq.,\,t}(\cdot) - p(\cdot \mid \mathbf{x}) \right\|_{\text{TV}} \tag{23}$$

$$\leq \sup_{h:\mathcal{Z}\to[-L/2,L/2]} \left| \mathbb{E}_{\eta_{\text{seq.},\,t}(\cdot)}\left[\, h \,\right] - \mathbb{E}_{p(\cdot|\mathbf{x})}\left[\, h \,\right] \right| \tag{24}$$

$$= L \left\| \eta_{\text{seq.},\,t}(\cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \tag{25}$$

$$\leq \frac{L}{N} \left(w^* - 1\right). \tag{26}$$

□

Theorem 1. *Assuming* $w^* = \sup_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})/q_{\boldsymbol{\lambda}_T}(\mathbf{z}) < \infty$ *for* $\forall \boldsymbol{\lambda}$ *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 at iteration t is bounded as*

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

*Proof of **Theorem** 1.* We denote the empirical distribution of the Markov-chain states at iteration t as

$$\eta_{\text{par.},\,t}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} K\left(\mathbf{z}_{t-1}^{(i)}, \mathbf{z}\right). \tag{27}$$

and consequently,

$$g_{\text{par.},t}(\boldsymbol{\lambda}) = \int s\left(\mathbf{z}; \boldsymbol{\lambda}\right)\, \eta_{\text{par.},\,t}(\mathbf{z})\, d\mathbf{z}. \tag{28}$$

Similarly with Section 3.2,

$$\left\| \eta_{\text{par.},\,t}(\mathbf{z}) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} = \left\| \frac{1}{N} \sum_{i=1}^{N} K\left(\mathbf{z}_{t-1}^{(i)}, \mathbf{z}\right) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \tag{29}$$

$$\leq \frac{1}{N} \sum_{i=1}^{N} \left\| K\left(\mathbf{z}_{t-1}^{(i)}, \cdot\right) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \qquad \text{(Triangle inequality)} \tag{30}$$

$$= \left\| K(\mathbf{z}_t, \cdot) - p\left(\cdot \mid \mathbf{x}\right) \right\|_{\text{TV}} \qquad \text{(Uniform ergodicity)} \tag{31}$$

$$\leq 1 - \frac{1}{w^*}. \tag{32}$$

And, finally the bias is given as

$$\begin{aligned}
\mathrm{bias}\left[\mathrm{g}_{\mathrm{par.,t}}\right] &\leq L \left\| \eta_{\mathrm{par.},\,t}(\cdot) - p(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} && (33) \\
&\leq L \sup_{h : \mathcal{Z} \rightarrow [-L/2, L/2]} \left| \mathbb{E}_{\eta_{\mathrm{par.},\,t}(\cdot)}\left[\, h \,\right] - \mathbb{E}_{p(\cdot \mid \mathbf{x})}\left[\, h \,\right] \right| && (34) \\
&= L \left\| \eta_{\mathrm{par.},\,t}(\cdot) - p\,(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} && (35) \\
&\leq L \left(1 - \frac{1}{w^*}\right) && (36)
\end{aligned}$$

□

Theorem 2. *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 $|\, s\,(\mathbf{z};\boldsymbol{\lambda})\,| \leq \frac{L}{2}$, the bias of the single state estimator at iteration t is bounded as*

$$\mathrm{Bias}\left[\mathrm{g}_{\mathrm{cis.,\,t}}\right] \leq L\,C \quad \textit{where} \quad C = \left(1 - \frac{N}{w^*}\right) < 1.$$

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

$$\eta_{\mathrm{cis.},\,t}(\mathbf{z}) = K\left(\mathbf{z}_{t-1}, \mathbf{z}\right), \tag{37}$$

and consequently,

$$\mathrm{g}_{\mathrm{cis},t}(\boldsymbol{\lambda}) = \int s\,(\mathbf{z};\boldsymbol{\lambda})\; \eta_{\mathrm{cis.},\,t}(\mathbf{z})\, d\mathbf{z}. \tag{38}$$

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

$$\left\| K^t\left(\mathbf{z}_{t-1}, \cdot\right) - p\,(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} \leq \left(1 - \frac{N-1}{2\,w^* + N - 2}\right)^t. \tag{39}$$

From this, the bound can be shown as

$$\begin{aligned}
\mathrm{bias}\left[\mathrm{g}_{\mathrm{cis.,t}}\right] &\leq \left\| \eta_{cis.,\,t}(\cdot) - p(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} && (40) \\
&\leq \sup_{h : \mathcal{Z} \rightarrow [-L/2, L/2]} \left| \mathbb{E}_{\eta_{\mathrm{cis.},\,t}(\cdot)}\left[\, h \,\right] - \mathbb{E}_{p(\cdot \mid \mathbf{x})}\left[\, h \,\right] \right| && (41) \\
&= L \left\| \eta_{\mathrm{cis.},\,t}(\cdot) - p\,(\cdot \mid \mathbf{x}) \right\|_{\mathrm{TV}} && (42) \\
&\leq L \left(1 - \frac{N-1}{2\,w^* + N - 2}\right) && (43) \\
&\leq L \left(1 - \frac{N}{w^*}\right) \qquad \text{(Monotonicity)} && (44)
\end{aligned}$$

given that $N > 2$. □

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

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

Proof of ***Proposition** 1.*

$$\begin{aligned}
&D_{\mathrm{KL}}(p\,(\cdot \mid \mathbf{x}) \parallel q_{\lambda}\,(\cdot)) && (45) \\
&= \int p\,(\mathbf{z} \mid \mathbf{x}) \log \frac{p\,(\mathbf{z} \mid \mathbf{x})}{q_{\lambda}\,(\mathbf{z})}\, d\mathbf{z} && (46) \\
&\leq \int p\,(\mathbf{z} \mid \mathbf{x}) \log M\, d\mathbf{z} && (47) \\
&= \log w^* && (48)
\end{aligned}$$

□

Theorem 3. *The variance of the sequential state estimator is*

*Proof of **Theorem** 3.* For notational convenience, let us define $g(\mathbf{z}) = s(\mathbf{z}; \boldsymbol{\lambda})$.

$$\mathbb{V}\left[\, g_{\text{seq.,t}} \,\right] = \mathbb{E}\left[\, \mathbb{V}\left[\, g \mid \mathbf{z}_T \,\right]\right] + \mathbb{V}\left[\, \mathbb{E}\left[\, g \mid \mathbf{z}_T \,\right]\right] \tag{49}$$

$$\begin{aligned}
&\mathbb{E}\left[\, \mathbb{V}\left[\, g \mid \mathbf{z}_T \,\right]\right] && (50)\\
&= \mathbb{E}\left[\, \frac{1}{N^2} \sum_{i=1}^{N} \mathbb{V}\left[\, g(\mathbf{z}_{T+i}) \mid \mathbf{z}_T \,\right] + \frac{2}{N^2} \sum_{i<j} \mathrm{Cov}\left(\, g(\mathbf{z}_{T+1}), g(\mathbf{z}_{T+i}) \mid \mathbf{z}_T \,\right) \right] && (51)\\
&= \frac{1}{N^2} \sum_{i=1}^{N} \mathbb{E}\left[\, \mathbb{V}\left[\, g(\mathbf{z}_{T+i}) \mid \mathbf{z}_T \,\right]\right] && (52)\\
&\qquad + \frac{2}{N^2} \sum_{i<j} \mathbb{E}\left[\, \mathrm{Cov}\left(\, g(\mathbf{z}_{T+1}), g(\mathbf{z}_{T+i}) \mid \mathbf{z}_T \,\right)\right] && (53)\\
&= \frac{1}{N}\sigma^2 + \frac{2}{N^2} \sum_{i<j} \mathbb{E}_{p(\mathbf{z}_T \mid \mathbf{x})}\left[\, \mathrm{Cov}\left(\, g\left(\mathbf{z}_{T+i}\right), g\left(\mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \,\right)\right] \qquad \text{(Stationarity)} && (54)\\
&= \frac{1}{N}\sigma^2 + \frac{2}{N^2} \sum_{i<j} \mathbb{E}_{p(\mathbf{z}_T \mid \mathbf{x})}\left[\, \mathrm{Cov}\left(\, g\left(\mathbf{z}_{T+i}\right), g\left(\mathbf{z}_{T+j}\right) \mid \mathbf{z}_T \,\right)\right] && (55)
\end{aligned}$$

□

Theorem 4. *The variance of the single mode estimator with a CIS kernel $\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right]$ is approximately bounded below such that*

$$\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right] \geq \frac{N^4 Z^4}{(w(\mathbf{z}_{t-1}) + NZ)^4} \mathbb{V}_{q_\lambda}\left[\, f_{IS} \,|\, \mathbf{z}_{t-1} \,\right], \tag{10}$$

where $Z = \mathbb{E}_{q_\lambda}[p(\mathbf{z}, \mathbf{x})/q_\lambda(\mathbf{z})] = \int p(\mathbf{z}, \mathbf{x}) d\mathbf{z}$ is the normalizing constant.

*Proof of **Theorem** 4.* By the law of total variance,

$$\begin{aligned}
\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right] &= \mathbb{V}\left[\, \mathbb{E}_{q_\lambda}\left[\, g \mid \mathbf{z}_{t-1} \,\right]\right] + \mathbb{E}\left[\, \mathbb{V}_{q_\lambda}\left[\, g \mid \mathbf{z}_{t-1} \,\right]\right] && (56)\\
&= \mathbb{V}\left[\, \mathbb{E}_{q_\lambda}\left[\, \mathbb{E}\left[\, g \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right]\right] \\
&\quad + \mathbb{E}\left[\, \mathbb{V}_{q_\lambda}\left[\, \mathbb{E}\left[\, g \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right] + \mathbb{E}_{q_\lambda}\left[\, \mathbb{V}\left[\, g \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right]\right] && (57)\\
&= \mathbb{V}\left[\, \mathbb{E}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right] + \mathbb{E}\left[\, \mathbb{V}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right] \\
&\quad + \mathbb{E}\left[\, \mathbb{E}_{q_\lambda}\left[\, \mathbb{V}\left[\, g \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right]\right] && (58)\\
&= \mathbb{V}\left[\, \mathbb{E}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right] + \mathbb{E}\left[\, \mathbb{V}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right] \\
&\quad + \mathbb{E}\left[\, \mathbb{E}_{q_\lambda}\left[\, \mathbb{E}\left[\, g_{\text{IS}}^2 \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right] + \left(\, \mathbb{E}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right]\right)^2 \,\right] && (59)
\end{aligned}$$

where g_{IS} denotes the importance sampling estimator given $\mathbf{z}^{(1:N)}$ and $\mathbf{z}^{(0)} = \mathbf{z}_{t-1}$ defined as $g_{\text{IS}} = \sum_{i=0}^{N} \frac{w\left(\mathbf{z}^{(i)}\right)}{\sum_{i=0}^{N} w\left(\mathbf{z}^{(i)}\right)} s\left(\mathbf{z}^{(i)}\right)$.

If we use Rao-Blackwellization, the last variance term vanishes, and we are left with

$$\mathbb{V}_{q_\lambda}\left[\, g_{\text{single}} \,\right] = \mathbb{V}\left[\, \mathbb{E}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right] + \mathbb{E}\left[\, \mathbb{V}_{q_\lambda}\left[\, g_{\text{IS}} \mid \mathbf{z}_{t-1} \,\right]\right]. \tag{60}$$

Therefore, in expectation,

Now,

$$\mathbb{E}_{q_\lambda}\left[\, g_{\mathrm{IS}} \mid \mathbf{z}_{t-1} \,\right] \tag{61}$$

$$= \mathbb{E}_{q_\lambda}\left[\frac{\sum_{i=0}^{N} w\left(\mathbf{z}^{(i)}\right) s\left(\mathbf{z}^{(i)}\right)}{\sum_{i=0}^{N} w\left(\mathbf{z}^{(i)}\right)} \middle| \mathbf{z}_{t-1} \right] \tag{62}$$

$$= \mathbb{E}_{q_\lambda}\left[\frac{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) s\left(\mathbf{z}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} + \frac{w\left(\mathbf{z}_{t-1}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} s\left(\mathbf{z}_{t-1}\right) \middle| \mathbf{z}_{t-1} \right] \tag{63}$$

$$= \mathbb{E}_{q_\lambda}\left[\frac{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) s\left(\mathbf{z}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} \middle| \mathbf{z}_{t-1} \right] \tag{64}$$

$$+ \mathbb{E}_{q_\lambda}\left[\frac{w\left(\mathbf{z}_{t-1}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} \middle| \mathbf{z}_{t-1} \right] s\left(\mathbf{z}_{t-1}\right) \tag{65}$$

$$= \mathbb{E}_{q_\lambda}\left[\frac{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) s\left(\mathbf{z}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} \middle| \mathbf{z}_{t-1} \right] + r\left(\mathbf{z}_{t-1}\right) s\left(\mathbf{z}_{t-1}\right) \tag{66}$$

$$= \mathbb{E}_{q_\lambda}\left[\frac{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) + w\left(\mathbf{z}_{t-1}\right)} \frac{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right) s\left(\mathbf{z}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\mathbf{z}^{(i)}\right)} \middle| \mathbf{z}_{t-1} \right] + r\left(\mathbf{z}_{t-1}\right) s\left(\mathbf{z}_{t-1}\right) \tag{67}$$

Write

$$\mathbb{V}_{q_\lambda}[f|\mathbf{z}_{t-1}] = \mathbb{V}_{q_\lambda}\left[\frac{\sum_{i=1}^{N} w(\mathbf{z}^{(i)})}{\sum_{i=0}^{N} w(\mathbf{z}^{(i)})} \frac{\sum_{i=1}^{N} w(\mathbf{z}^{(i)}) f(\mathbf{z}^{(i)})}{\sum_{i=1}^{N} w(\mathbf{z}^{(i)})} + \frac{w(\mathbf{z}_{t-1})}{\sum_{i=0}^{N} w(\mathbf{z}^{(i)})} f(\mathbf{z}_{t-1}) \middle| \mathbf{z}_{t-1} \right]. \tag{68}$$

Note that if $a > 0$, then we can approximate the function $\sum_{i=1}^{N} x_i/(a + \sum_{i=1}^{N} x_i)$ using the first-order Taylor series expansion about $(Z, \dots, Z)$ by

$$\frac{\sum_{i=1}^{N} x_i}{a + \sum_{i=1}^{N} x_i} \approx \frac{NZ}{a + NZ} + \sum_{i=1}^{N} \frac{a}{(a+NZ)^2}(x_i - Z).$$

Hence, given $\mathbf{z}_{t-1}$, we approximate $\sum_{i=1}^{N} w(\mathbf{z}^{(i)}) / \sum_{i=0}^{N} w(\mathbf{z}^{(i)})$ by

$$\frac{\sum_{i=1}^{N} w(\mathbf{z}^{(i)})}{\sum_{i=0}^{N} w(\mathbf{z}^{(i)})} \approx \frac{NZ}{w(\mathbf{z}_{t-1}) + NZ} + \sum_{i=1}^{N} \frac{w(\mathbf{z}_{t-1})}{(w(\mathbf{z}_{t-1}) + NZ)^2}(w(\mathbf{z}^{(i)}) - Z) \tag{69}$$

$$= \frac{N^2 Z^2 + w(\mathbf{z}_{t-1}) \sum_{i=1}^{N} w(\mathbf{z}^{(i)})}{(w(\mathbf{z}_{t-1}) + NZ)^2} \tag{70}$$

so that

$$\mathbb{V}_{q_\lambda}[f|\mathbf{z}_{t-1}] \approx \mathbb{V}_{q_\lambda}\left[\frac{N^2 Z^2}{(w(\mathbf{z}_{t-1}) + NZ)^2} f_{IS} + \frac{w(\mathbf{z}_{t-1})}{(w(\mathbf{z}_{t-1}) + NZ)^2} \sum_{i=1}^{N} w(\mathbf{z}^{(i)}) f(\mathbf{z}^{(i)}) \right. \tag{71}$$

$$\left. + \frac{w(\mathbf{z}_{t-1})}{\sum_{i=0}^{N} w(\mathbf{z}^{(i)})} f(\mathbf{z}_{t-1}) \middle| \mathbf{z}_{t-1} \right]. \tag{72}$$

Observe that $\sum_{i=1}^{N} w(\mathbf{z}^{(i)}) f(\mathbf{z}^{(i)}) = O(N)$ since $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}\}$ are independent and identically distributed and that $w(\mathbf{z}_{t-1}) f(\mathbf{z}_{t-1}) / \sum_{i=0}^{N} w(\mathbf{z}^{(i)}) = o(N)$. Combining these, we obtain

$$\mathbb{V}_{q_\lambda}\left[\, f \mid \mathbf{z}_{t-1}, \mathbf{z}^{(1:N)} \,\right] \approx \frac{N^4 Z^4}{(w(\mathbf{z}_{t-1}) + NZ)^4} \mathbb{V}_{q_\lambda}\left[\, f_{\mathrm{IS}} \,|\, \mathbf{z}_{t-1} \,\right], \tag{73}$$

as was to be shown. □