MARKOV CHAIN SCORE ASCENT: A Unifying Framework of Variational Inference with Markovian Gradients

Anonymous Author(s)
Affiliation
Address
email

Abstract

Minimizing the inclusive Kullback-Leibler (KL) divergence with stochastic gradient descent (SGD) is challenging since its gradient is defined as an integral over the posterior. Recently, multiple methods have been proposed to run SGD with *biased* gradient estimates obtained from a Markov chain. This paper provides the first non-asymptotic convergence analysis of these methods by establishing their mixing rate and gradient variance. To do this, we demonstrate that these methods–which we collectively refer to as Markov chain score ascent (MCSA) methods–can be cast as special cases of the Markov chain gradient descent framework. Furthermore, by leveraging this new understanding, we develop a novel MCSA scheme, *parallel* MCSA (pMCSA), that achieves a tighter bound on the gradient variance. We demonstrate that this improved theoretical result translates to superior empirical performance.

1 Introduction

Bayesian inference aims to analyze the posterior distribution of an unknown latent variable $\mathbf{z}$ from which data $\mathbf{x}$ is observed. By assuming a model $p(\mathbf{x} \mid \mathbf{z})$, the posterior $\pi(\mathbf{z})$ is given by Bayes' rule such that $\pi(\mathbf{z}) \propto p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})$ where $p(\mathbf{z})$ represents our prior belief on $\mathbf{z}$. Instead of working directly with π, variational inference (VI, Blei *et al.* 2017) seeks a *variational approximation* $q(\mathbf{z}; \boldsymbol{\lambda})$ that is the most similar to π according to a discrepancy measure $d(\pi,\ q(\cdot; \boldsymbol{\lambda}))$.

The apparent importance of choosing the right discrepancy measure has led to a quest spanning a decade (Bamler *et al.*, 2017; Dieng *et al.*, 2017; Geffner & Domke, 2021b; Hernandez-Lobato *et al.*, 2016; Li & Turner, 2016; Regli & Silva, 2018; Ruiz & Titsias, 2019; Salimans *et al.*, 2015; Wan *et al.*, 2020; Wang *et al.*, 2018; Zhang *et al.*, 2021). So far, the exclusive (or reverse, backward) Kullback-Leibler (KL) divergence $d_{\mathrm{KL}}(q(\cdot; \boldsymbol{\lambda}) \parallel \pi)$ has seen "exclusive" use, partly because it is defined as an integral over $q(\mathbf{z}; \boldsymbol{\lambda})$, which can be approximated efficiently. In contrast, the *inclusive* (or forward) KL is defined as an integral over π as

$$d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda})) = \int \pi(\mathbf{z}) \log \frac{\pi(\mathbf{z})}{q(\mathbf{z}; \boldsymbol{\lambda})}\, d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim \pi(\cdot)} \left[\log \frac{\pi(\mathbf{z})}{q(\mathbf{z}; \boldsymbol{\lambda})} \right].$$

Since our goal is to approximate π with $q(\cdot; \boldsymbol{\lambda})$ but the inclusive KL involves an integral over π, we end up facing a chicken-and-egg problem. Despite this challenge, the inclusive KL has consistently drawn attention due to its statistical properties, such as better uncertainty estimates due to its mass covering property (MacKay, 2001; Minka, 2005; Trippe & Turner, 2017).

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Recently, Naesseth *et al.* (2020); Ou & Song (2020) have respectively proposed Markovian score climbing (MSC) and joint stochastic approximation (JSA). These methods minimize the inclusive KL using stochastic gradient descent (SGD, Robbins & Monro 1951), where the gradients are estimated using a Markov chain. The Markov chain kernel $K_{\boldsymbol{\lambda}_t}(\mathbf{z}_t, \cdot)$ is π-invariant (Robert & Casella, 2004) and is chosen such that it directly takes advantage of the current variational approximation $q(\cdot; \boldsymbol{\lambda}_t)$. Thus, the quality of the gradients improves over time as the KL divergence decreases. Still, the gradients are non-asymptotically biased and Markovian across adjacent iterations, which sharply contrasts MSC and JSA from classical black-box VI (Kucukelbir *et al.*, 2017; Ranganath *et al.*, 2014), where the gradients are unbiased and independent. While Naesseth *et al.* (2020) has shown the convergence of MSC through the work of Gu & Kong (1998), this result is only asymptotic and does not provide practical insight into the performance of MSC.

In this paper, we address these theoretical gaps by casting MSC and JSA into a general framework we call Markov chain score ascent (MCSA), which we show is a special case of Markov chain gradient descent (MCGD, Duchi *et al.* 2012). This enables the application of the non-asymptotic convergence results of MCGD (Debavelaere *et al.*, 2021; Doan *et al.*, 2020a,b; Duchi *et al.*, 2012; Karimi *et al.*, 2019; Sun *et al.*, 2018; Xiong *et al.*, 2021). For MCGD methods, the fundamental properties affecting the convergence rate are the ergodic convergence rate (ρ) of the MCMC kernel and the gradient variance (G). We analyze ρ and G of MSC and JSA, enabling their practical comparison given a fixed computational budget (N). Furthermore, based on the recent insight that the mixing rate does not affect the convergence rate of MCGD Doan *et al.* (2020a,b), we propose a novel scheme, parallel MCSA (pMCSA), which achieves lower variance by trading off the mixing rate. We verify our theoretical analysis through numerical simulations and compare MSC, JSA, and pMCSA on general Bayesian inference problems. Our experiments show that our proposed method outperforms previous MCSA approaches.

Contribution Summary

1. **Section 4:** We provide the first non-asymptotic theoretical analysis of two recently proposed inclusive KL minimization methods, MSC **(Theorem 1)** and JSA **(Theorem 2)**.
2. **Section 3:** To do this, we show that both methods can be viewed as what we call "Markov chain score ascent" (MCSA) methods, which are a special case of MCGD **(Proposition 1)**.
3. **Section 5:** In light of this, we develop a novel MCSA method which we call parallel MCSA (pMCSA) that achieves lower gradient variance **(Theorem 3)**.
4. **Section 6:** We demonstrate that the improved theoretical performance of pMCSA translates to superior empirical performance across a variety of Bayesian inference tasks.

2 Background

2.1 Inclusive Kullback-Leibler Minimization with Stochastic Gradients

VI with SGD The goal of VI is to find the optimal variational parameter $\boldsymbol{\lambda}$ identifying $q(\cdot; \boldsymbol{\lambda}) \in \mathcal{Q}$ that minimizes some discrepancy measure $D(\pi, q(\cdot; \boldsymbol{\lambda}))$. A typical way to perform VI is to use stochastic gradient descent (SGD, Robbins & Monro 1951), provided that *unbiased* gradient estimates of the optimization target $\mathbf{g}(\boldsymbol{\lambda})$ are available such that we can repeat the update

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

where $\gamma_1, \dots, \gamma_T$ is a step-size schedule.

Inclusive KL Minimization with SGD For inclusive KL minimization, $\mathbf{g}$ should be set as

$$\mathbf{g}(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda})) = \nabla_{\boldsymbol{\lambda}} \mathbb{H}[\pi, q(\cdot; \boldsymbol{\lambda})] = -\mathbb{E}_{\mathbf{z} \sim \pi(\cdot)}[\, \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}) \,]$$

where $\mathbb{H}[\pi, q(\cdot; \boldsymbol{\lambda})]$ is the cross-entropy between π and $q(\cdot; \boldsymbol{\lambda})$, which shows the connection with cross-entropy methods (de Boer *et al.*, 2005), and $\mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}) = \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z}; \boldsymbol{\lambda})$ is known as the *score gradient*. Since inclusive KL minimization with SGD is equivalent to ascending towards the direction of the score, Naesseth *et al.* (2020) coined the term score climbing. To better conform with the optimization literature, we instead call this approach *score ascent* as in gradient ascent.

2.2 Markov Chain Gradient Descent

Overview of MCGD Markov chain gradient descent (MCGD, Duchi *et al.* 2012; Sun *et al.* 2018) is a family of algorithms that optimize a function f defined as $f(\boldsymbol{\lambda}) = \int f(\boldsymbol{\lambda}, \boldsymbol{\eta})\, \Pi(d\boldsymbol{\eta})$ where $\Pi(d\boldsymbol{\eta})$ is a probability measure. MCGD repeats the steps

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \gamma_t\, \mathbf{g}(\boldsymbol{\lambda}_t, \boldsymbol{\eta}_t), \quad \boldsymbol{\eta}_t \sim P_{\boldsymbol{\lambda}_{t-1}}(\boldsymbol{\eta}_{t-1}, \cdot) \tag{1}$$

where $P_{\boldsymbol{\lambda}_{t-1}}$ is a Π-invariant Markov chain kernel that may depend on $\boldsymbol{\lambda}_{t-1}$. The noise of the gradient is Markovian and non-asymptotically biased, departing from vanilla SGD. Non-asymptotic convergence of this general algorithm has recently started to gather attention as by Debavelaere *et al.* (2021); Doan *et al.* (2020a,b); Duchi *et al.* (2012); Karimi *et al.* (2019); Sun *et al.* (2018).

Applications of MCGD MCGD encompasses an extensive range of problems, including distributed optimization (Ram *et al.*, 2009), reinforcement learning (Doan *et al.*, 2020a; Tadić & Doucet, 2017; Xiong *et al.*, 2021), and expectation-minimization (Karimi *et al.*, 2019), to name a few. This paper extends this list with inclusive KL VI through the MCSA framework.

3 Markov Chain Score Ascent

First, we develop Markov chain score ascent (MCSA), a framework for inclusive KL minimization with MCGD. This framework will establish the connection between MSC/JSA and MCGD.

3.1 Markov Chain Score Ascent as a Special Case of Markov Chain Gradient Descent

As shown in Equation (1), the basic ingredients of MCGD are the target function $f(\boldsymbol{\lambda}, \eta)$, the gradient estimator $\mathbf{g}(\boldsymbol{\lambda}, \eta)$, and the Markov chain kernel $P_{\boldsymbol{\lambda}}(\eta, \cdot)$. Obtaining MCSA from MCGD boils down to designing $\mathbf{g}$ and $P_{\boldsymbol{\lambda}}$ such that $f(\boldsymbol{\lambda}) = d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda}))$. The following proposition provides sufficient conditions on $\mathbf{g}$ and $P_{\boldsymbol{\lambda}}$ to achieve this goal.

Proposition 1. Let $\boldsymbol{\eta} = \left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N)}\right)$ and a Markov chain kernel $P_{\boldsymbol{\lambda}}(\boldsymbol{\eta}, \cdot)$ be Π-invariant where Π is defined as

$$\Pi(\boldsymbol{\eta}) = \pi\left(\mathbf{z}^{(1)}\right) \pi\left(\mathbf{z}^{(2)}\right) \times \ldots \times \pi\left(\mathbf{z}^{(N)}\right).$$

Then, by defining the target function f and the gradient estimator $\mathbf{g}$ to be

$$f(\boldsymbol{\lambda}, \boldsymbol{\eta}) = -\frac{1}{N} \sum_{n=1}^{N} \log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) - \mathbb{H}[\pi], \quad \text{and} \quad \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) = -\frac{1}{N} \sum_{n=1}^{N} \mathbf{s}\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)$$

where $\mathbb{H}[\pi]$ is the entropy of π, MCGD results in inclusive KL minimization as

$$\mathbb{E}_{\Pi}[f(\boldsymbol{\lambda}, \boldsymbol{\eta})] = d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda})), \quad \text{and} \quad \mathbb{E}_{\Pi}[\mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta})] = \nabla_{\boldsymbol{\lambda}} d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda})).$$

Proof. See page 18 of the *supplementary material*.

This simple connection between MCGD and VI paves the way toward the non-asymptotic analysis of JSA and MSC. Note that N here can be regarded as the computational budget of each MCGD iteration since the cost of **(i)** generating the Markov chain samples $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$ and **(ii)** computing the gradient $\mathbf{g}$ will linearly increase with N.

In addition, the MCGD framework often assumes P to be geometrically ergodic. An exception is the analysis of Debavelaere *et al.* (2021) where they work with polynomially ergodic kernels.

Assumption 1. *(Markov chain kernel) The Markov chain kernel P is geometrically ergodic as*

$$d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^{n}(\boldsymbol{\eta}, \cdot), \Pi\right) \leq C\, \rho^{n}$$

for some positive constant C.

3.2 Non-Asymptotic Convergence of Markov Chain Score Ascent

Non-Asymptotic Convergence Through Proposition 1 and Assumption 1 and some technical assumptions on the objective function, we can apply the existing convergence results of MCGD to MCSA. Table 1 provides a list of relevant results. Apart from properties of the objective function (such as Lipschitz smoothness), the convergence rates are stated in terms of the gradient bound G, kernel mixing rate ρ, and the number of MCGD iteration T. We focus on G and ρ as they are closely related to the design choices of different MCSA algorithms.

Table 1: Convergence Rates of MCGD Algorithms

Algorithm	Stepsize Rule	Gradient Assumption	Rate	Reference
Mirror Descent[1]	$\gamma_t = \gamma/\sqrt{t}$	$\mathbb{E}\left[\, \| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \mid \mathcal{F}_t \right] < G^2$	$\mathcal{O}\left(\frac{G^2 \log T}{\log \rho^{-1} \sqrt{T}}\right)$	Duchi *et al.* (2012) Corollary 3.5
SGD-Nesterov[2]	$\gamma_t = 2/(t+1)$ $\beta_t = \frac{1}{2L\sqrt{t+1}}$	$\|\mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta})\|_2 < G$	$\mathcal{O}\left(\frac{G^2 \log T}{\sqrt{T}}\right)$	Doan *et al.* (2020a) Theorem 2
SGD[3]	$\gamma_t = \gamma/t$ $\gamma = \min\{1/2L, 2L/\mu\}$	$\| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_* < G(\| \boldsymbol{\lambda} \|_2 + 1)$	$\mathcal{O}\left(\frac{G^2 \log T}{T}\right)$	Doan *et al.* (2020b) Theorem 1,2

Notation: [1]$\mathcal{F}_t$ is the σ-field formed by all the iterates $\boldsymbol{\eta}_t, \boldsymbol{\lambda}_t$ up to the tth MCGD iteration and $\| \mathbf{x} \|_*$ is the dual norm such that $\| \mathbf{x} \|_* = \sup_{\| \mathbf{z} \| \leq 1} \langle \mathbf{x}, \mathbf{z} \rangle$. [2]$\beta_t$ is the stepsize of the momentum. [2,3]L is the Lipschitz smoothness constant. [3]μ is the strong convexity constant.

Convergence and the Mixing Rate ρ Duchi *et al.* (2012) was the first to provide an analysis of the general MCGD setting. Their convergence rate is dependent on the mixing rate through the $1/\log \rho^{-1}$ term. For MCSA, this results in an overly conservative rate since, on challenging problems, mixing can be slow such that $\rho \approx 1$. Fortunately, Doan *et al.* (2020a,b) have recently shown that it is possible to obtain a rate independent of the mixing rate ρ. For example, in the result of Doan *et al.* (2020b), the influence of ρ decreases in a rate of $\mathcal{O}(1/T^2)$. This observation is critical since it implies that *trading "lower gradient variance" with a "slower mixing rate" could be profitable*. We exploit this observation in our novel MCSA scheme in Section 5.

Gradient Bound G Except for Doan *et al.* (2020b), most results assume that the gradient is bounded for $\forall \boldsymbol{\eta}, \boldsymbol{\lambda}$ as $\|\mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta})\| < G$. Admittedly, this condition is strong, but it is similar to the bounded variance assumption $\mathbb{E}[\| \mathbf{g} \|^2] < G^2$ used in vanilla SGD, which is also known to be strong as it contradicts strong convexity (Nguyen *et al.*, 2018). Nonetheless, assuming G can have practical benefits beyond theoretical settings. For example, Geffner & Domke (2020) use G to compare the performance different VI gradient estimators. In a similar spirit, we will obtain the gradient bound G of different MCSA algorithms and compare their theoretical performance.

4 Demystifying Prior Markov Chain Score Ascent Methods

In this section, we will show that MSC and JSA both qualify as MCSA methods. Furthermore, we establish **(i)** the mixing rate of their implicitly defined kernel P and **(ii)** the upper bound on their gradient variance. This will provide insight into their practical non-asymptotic performance.

4.1 Technical Assumptions

To cast previous methods into MCSA, we need some technical assumptions.

Assumption 2. *(Bounded importance weight) The importance weight ratio $w(\mathbf{z}) = \pi(\mathbf{z})/q(\mathbf{z}; \boldsymbol{\lambda})$ is bounded by some finite constant as $w^* < \infty$ for all $\boldsymbol{\lambda} \in \Lambda$ such that $\rho = (1 - 1/w^*) < 1$.*

This assumption is necessary to ensure Assumption 1, and can be practically ensured by using a variational family with heavy tails (Domke & Sheldon, 2018) or using a defensive mixture (Hesterberg, 1995; Holden *et al.*, 2009) as

$$q_{\text{def.}}(\mathbf{z}; \boldsymbol{\lambda}) = w\, q(\mathbf{z}; \boldsymbol{\lambda}) + (1 - w)\, \nu(\mathbf{z})$$

where $0 < w < 1$ and $\nu(\cdot)$ is a heavy tailed distribution such that $\sup_{\mathbf{z} \in \mathcal{Z}} \pi(\mathbf{z})/\nu(\mathbf{z}) < \infty$. Note that $q_{\text{def.}}$ is only used in the Markov chain kernels and $q(\cdot; \boldsymbol{\lambda}^*)$ is still the output of the VI procedure. While these tricks help escape slowly mixing regions, this benefit quickly vanishes as we converge. Therefore, ensuring Assumption 2 seems unnecessary in practice unless we absolutely care about ergodicity (*e.g.* adaptive MCMC, Brofos *et al.* 2022; Holden *et al.* 2009).

Model (Variational Family) Misspecification and w^* Note that w^* is bounded below exponentially by the inclusive KL as shown in Proposition 2. Therefore, w^* will be large **(i)** in the initial steps of VI and **(ii)** under model (variational family) misspecification.

Assumption 3. *(Bounded Score) The score gradient is bounded for $\forall \boldsymbol{\lambda} \in \Lambda$ and $\forall \mathbf{z} \in \mathcal{Z}$ such that $\| \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}) \|_* \leq L$ for some finite constant $L > 0$.*

Although this assumption is strong, it enables us to compare the gradient variance of MCSA methods. We empirically justify the bounds obtained using Assumption 3 in Section 6.2.

4.2 Markovian Score Climbing

MSC (Algorithm 3) is a simple instance of MCSA where $\eta_t = \mathbf{z}_t$ and $P_{\lambda_t} = K_{\lambda_t}$, where K_{λ_t} is the conditional importance sampling (CIS) kernel (originally proposed by Andrieu *et al.* (2018)) where the proposals are generated from $q(\cdot; \boldsymbol{\lambda}_t)$. Although MSC uses only a single sample for the Markov chain, the CIS kernel internally uses N proposals to generate a single sample. Therefore, N in MSC has a different meaning, but it still indicates the computational budget.

Theorem 1. MSC (Naesseth *et al.*, 2020) is obtained by defining

$$P_\lambda^k(\boldsymbol{\eta}, d\boldsymbol{\eta}') = K_\lambda^k(\mathbf{z}, d\mathbf{z}')$$

with $\boldsymbol{\eta}_t = \mathbf{z}_t$ where $K_\lambda(\mathbf{z}, \cdot)$ is the CIS kernel with $q_{def.}(\cdot; \boldsymbol{\lambda})$ as its proposal distribution. Then, given Assumption 2 and 3, the mixing rate and the gradient bounds are given as

$$d_{\mathrm{TV}}\left(P_\lambda^k(\boldsymbol{\eta}, \cdot), \Pi\right) \leq \left(1 - \frac{N-1}{2w^* + N - 2}\right)^k \quad \text{and} \quad \mathbb{E}\left[\, \| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \,\middle|\, \mathcal{F}_t \,\right] \leq L^2,$$

where $w^* = \sup_{\mathbf{z}} \pi(\mathbf{z}) / q_{\text{def.}}(\mathbf{z}; \boldsymbol{\lambda})$.

Proof. See page 20 of the *supplementary material*.

Discussion Theorem 1 shows that the gradient variance of MSA is insensitive to N. Although the mixing rate does improve with N, when w^* is large due to model misspecification and lack of convergence (see the discussion in Section 4.1), this will be marginal. Overall, *the performance of MSC cannot be improved by increasing the computational budget N*. Even under stationarity, the variance of $\boldsymbol{\eta}$ is equal to that of a *single* posterior sample at best.

4.3 Joint Stochastic Approximation

JSA (Algorithm 4) was proposed for deep generative models where the likelihood factorizes into each datapoint, for which subsampling can be used through a random-scan version of the independent Metropolis-Hastings (IMH, Hastings 1970) kernel. Instead, we consider the general version of JSA with the vanilla IMH kernel since it can be used with any type of likelihood. At each MCGD step, JSA performs multiple Markov chain transitions and estimates the gradient by averaging all the intermediate states, which is closer to traditional MCMC estimation.

Independent Metropolis-Hastings Similarly to MSC, the IMH kernel in JSA generates proposals from $q(\cdot; \boldsymbol{\lambda}_t)$. To show the geometric ergodicity of the implicit kernel P, we utilize the geometric convergence rate of IMH kernels provided by Mengersen & Tweedie (1996, Theorem 2.1) and Wang (2020). Furthermore, to derive an upper bound on the gradient variance, we use the exact n-step marginal IMH kernel derived by Smith & Tierney (1996) as

$$K_\lambda^n(\mathbf{z}, d\mathbf{z}') = T_n\left(w(\mathbf{z}) \vee w(\mathbf{z}')\right) \pi(\mathbf{z}')\, d\mathbf{z}' + \lambda^n(w(\mathbf{z}))\, \delta_{\mathbf{z}}(d\mathbf{z}') \tag{2}$$

where $w(\mathbf{z}) = \pi(\mathbf{z}) / q_{\text{def.}}(\mathbf{z}; \boldsymbol{\lambda})$, $x \vee y = \max(x, y)$, and for $R(w) = \{\, \mathbf{z}' \mid w(\mathbf{z}') \leq w \,\}$,

$$T_n(w) = \int_w^\infty \frac{n}{v^2} \lambda^{n-1}(v)\, dv, \quad \text{and} \quad \lambda(w) = \int_{R(w)} \left(1 - \frac{w(\mathbf{z}')}{w}\right) \pi(d\mathbf{z}'). \tag{3}$$

Theorem 2. JSA (Ou & Song, 2020) is obtained by defining

$$P_\lambda^k(\boldsymbol{\eta}, d\boldsymbol{\eta}') = K_\lambda^{N(k-1)+1}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{(1)}\right) K_\lambda^{N(k-1)+2}\left(\mathbf{z}^{(2)}, d\mathbf{z}'^{(2)}\right) \cdot \ldots \cdot K_\lambda^{N(k-1)+N}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{(N)}\right)$$

with $\boldsymbol{\eta}_t = \left(\mathbf{z}_t^{(1)}, \mathbf{z}_t^{(2)}, \ldots, \mathbf{z}_t^{(N)}\right)$. Then, given Assumption 2 and 3, the mixing rate and the gradient variance bounds are

$$d_{\mathrm{TV}}\left(P_\lambda^k(\boldsymbol{\eta}, \cdot), \Pi\right) \leq C(\rho, N)\, \rho^{kN} \quad \text{and} \quad \mathbb{E}\left[\, \| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \,\middle|\, \mathcal{F}_t \,\right] \leq L^2 \left[\, \frac{1}{2} + \frac{3}{2}\frac{1}{N} + \mathcal{O}\left(1/w^*\right) \right],$$

where $w^* = \sup_{\mathbf{z}} \pi(\mathbf{z}) / q_{\text{def.}}(\mathbf{z}; \boldsymbol{\lambda})$ and $C(\rho, N) > 0$ is a finite constant depending on ρ and N.

Proof. See page 21 of the *supplementary material*.

Discussion According to Theorem 2, JSA benefits from increasing N both in terms of faster mixing rate and lower gradient variance. However, similarly to MSC, under lack of convergence and model misspecification (large w^*), the improvement becomes marginal. More importantly, *in the large w^* regime, the variance reduction is limited by the constant $1/2$ term*. This is because a large w^* results in more Metropolis-Hastings rejections, increasing the correlation between the N samples. This inefficiency persists even under stationarity unless $\pi(\cdot) \in \mathcal{Q}$ and $q(\cdot; \boldsymbol{\lambda}) = \pi(\cdot)$.

5 Parallel Markov Chain Score Ascent

Our analysis in Section 4 suggests that the statistical performance of both MSC and JSA are heavily affected by model specification and the state of convergence through w^*. This affects both the mixing rate ρ and the gradient variance G. Furthermore, a large w^* abolishes our ability to counterbalance the inefficiency by increasing the computational budget N. However, ρ and G do not impact convergence equally. In particular, we noted in Section 3.1 that recent results on MCGD suggest that MCSA depends more on the gradient variance than the mixing rate. We turn to leverage this understanding to overcome the limitations of previous MCSA methods.

5.1 Parallel Markov Chain Score Ascent

We propose a novel scheme, *parallel Markov chain score ascent* (pMCSA, Algorithm 5), that embraces a slower mixing rate in order to consistently achieve an $\mathcal{O}(1/N)$ variance reduction, even on challenging problems with in a large w^*,

Algorithm Description Unlike JSA, that uses N *sequential* Markov chain states, pMCSA operates N parallel Markov chains. To maintain a similar per-iteration cost with JSA, it performs only a single Markov chain transition for each chain. Since the chains are independent, the Metropolis-Hastings rejections do not affect the variance of pMCSA.

Theorem 3. pMCSA, our proposed scheme, is obtained by setting

$$P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, d\boldsymbol{\eta}'\right) = K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{(1)}\right) K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(2)}, d\mathbf{z}'^{(2)}\right) \cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{(N)}\right)$$

with $\boldsymbol{\eta} = \left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N)}\right)$. Then, given Assumption 2 and 3, the mixing rate and the gradient variance bounds are

$$d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, \cdot\right), \Pi\right) \leq C\left(N\right) \rho^{k} \quad \text{and} \quad \mathbb{E}\left[\, \| \mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right) \|_{*}^{2} \,\middle|\, \mathcal{F}_{t} \,\right] \leq L^{2}\left[\, \frac{1}{N} + \frac{1}{N}\left(1 - \frac{1}{w^*}\right) \,\right],$$

where $w^* = \sup_{\mathbf{z}} \pi\left(\mathbf{z}\right) / q_{\text{def.}}\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ and C is some positive constant depending on N.

Proof. See page 23 of the *supplementary material*.

Discussion Unlike JSA and MSC, the variance reduction rate of pMCSA is independent of w^*. Therefore, it should perform significantly better on challenging practical problems. If we consider the rate of Duchi *et al.* (2012), the combined rate is constant with respect to N since it cancels out. In practice, however, we observe that increasing N accelerates convergence quite dramatically. Therefore, the mixing rate independent convergence rates by Doan *et al.* (2020a,b) appears to better reflect practical performance. This is because **(i)** the mixing rate ρ is a conservative *global* bound and **(ii)** the mixing rate will improve naturally as MCSA converges.

5.2 Computational Cost Comparison

The three schemes using the CIS and IMH kernels have different computational costs depending on N as organized in Table 2.

Table 2: Computational Costs

	Applying $P_{\boldsymbol{\lambda}}$			Estimating $\mathbf{g}$	
	$p\left(\mathbf{z}, \mathbf{x}\right)$ # Eval.	$q\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ # Eval.	$q\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ # Samples	$p\left(\mathbf{z}, \mathbf{x}\right)$ # Grad.	$q\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ # Grad.
ELBO	0	0	N	N	N
MSC	$N-1$	N	$N-1$	0	1^1 or N^2
JSA	N	$N+1$	N	0	N
pMCSA	N	$2N$	N	0	N

Note: We assume that the variables are cached as much as possible. [1]Vanilla CIS kernel. [2]Rao-Blackwellized CIS kernel.

Cost of Sampling Proposals For the CIS kernel used by MSC, N controls the number of internal proposals sampled from $q\left(\cdot; \boldsymbol{\lambda}\right)$. For JSA and pMCSA, the IMH kernel only uses a single sample from $q\left(\cdot; \boldsymbol{\lambda}\right)$, but applies the kernel N times. On the other hand, pMCSA needs twice more evaluations of $q\left(\cdot; \boldsymbol{\lambda}\right)$. However, this added cost is minimal since it is dominated by that of evaluating $p\left(\mathbf{z}, \mathbf{x}\right)$.

Cost of Estimating the Score When estimating the score, MSC computes $\nabla_{\boldsymbol{\lambda}} \log q\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ only once, while JSA and our proposed scheme compute it N times. However, Naesseth *et al.* (2020) also discuss a Rao-Blackwellized version of the CIS kernel, which also computes the score N times. Lastly, notice that MCSA methods do not need to differentiate through the likelihood $p\left(\mathbf{z}, \mathbf{x}\right)$, unlike ELBO maximization, making its per-iteration cost significantly cheaper.

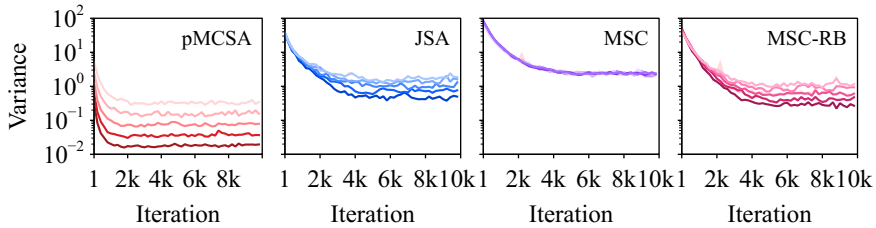

Figure 1: **Gradient variance versus iteration and computational budget (N). pMCSA not only achieves the least gradient variance, but its variance also scales better with N.** The colors range from light ($N = 2^3$) to dark ($N = 2^7$) representing the computational budgets $N \in [2^3, 2^4, 2^5, 2^6, 2^7]$. The target distribution is a 30-D multivariate Gaussian with $\nu = 150$. The error bands are the 80% quantiles obtained from 8 replications.

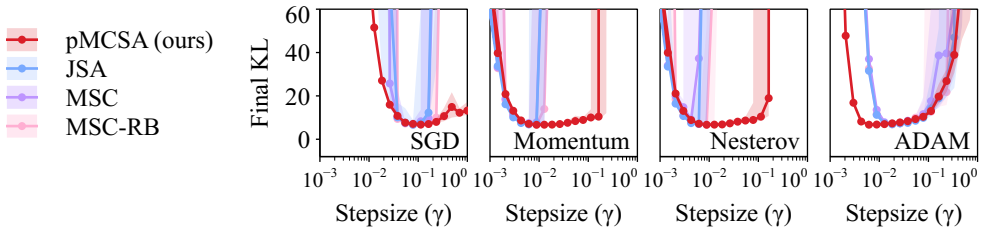

Figure 2: **Optimizer stepsize (γ) versus final KL. pMCSA is the least sensitive to optimizer hyperparameters and results in stable convergence.** The final KL is obtained at the 10^4th iteration. The target distribution is a 100-D Gaussian with $\nu = 500$. The error bands are the 80% quantiles, while the solid lines are the median of 20 replications.

6 Evaluations

6.1 Experimental Setup

Implementation For the realistic experiments, we implemented MCSA methods on top of the Turing (Ge *et al.*, 2018) probabilistic programming framework.For the variational family, we use diagonal multivariate Gaussians with the support transformation of Kucukelbir *et al.* (2017). We use the ADAM optimizer by Kingma & Ba (2015) with a stepsize of 0.01 in all experiments. The budget is set to $N = 10$ for all experiments unless specified.

Baselines We compare **(i) pMCSA** (ours, Section 5), **(ii) JSA** (Ou & Song, 2020), **(iii) MSC** (Naesseth *et al.*, 2020), **(iv)** MSC with with Rao-Blackwellization (**MSC-RB**, Naesseth *et al.* 2020), and **(v)** evidence lower-bound maximization (**ELBO**, Kucukelbir *et al.* 2017; Ranganath *et al.* 2014) with the path derivative estimator (Roeder *et al.*, 2017).

6.2 Simulations

Setup First, we verify our theoretical analysis on multivariate Gaussians with full-rank covariances sampled from Wishart distribution with ν degrees of freedom (values of ν are in the figure captions). This problem is challenging since an IMH (used by pMCSA, JSA) or CIS (used by MSC, MSC-RB) kernel with a diagonal Gaussian proposal will mix slowly due to a large w^*.

Gradient Variance We evaluate our theoretical analysis of the gradient variance. The variance is estimated from 512 independent Markov chains using the parameters generated by the main MCSA procedure. The estimated variances are shown in Figure 1. We make the following observations: **(i)** pMCSA has the lowest variance overall, and it consistently benefits from increasing N. **(ii)** MSC does not benefit from increasing N whatsoever. **(iii)** JSA does not benefit from increasing N until $q(\cdot; \boldsymbol{\lambda})$ has sufficiently converged (when w^* has become small). These results confirm our theoretical results in Sections 4 and 5.

Robustness Against Optimizers Since the convergence of most sophisticated SGD optimizers has yet to be established for MCGD, we empirically investigate their effectiveness. The results using SGD (Bottou *et al.*, 2018; Robbins & Monro, 1951), Momentum (Polyak, 1964), Nesterov (Nesterov, 1983), ADAM (Kingma & Ba, 2015), and varying stepsizes are shown in Figure 2. Clearly, pMCSA successfully converges for the broadest variety of optimizer settings. Overall, most MCSA methods seem to be the most stable with ADAM, which points out that establishing the convergence of ADAM for MCGD will be a promising direction for future works.

Table 3: Test Log Predictive Density on **Bayesian Neural Network Regression**

	D_{λ}	$D_{\mathbf{x}}$	N_{train}	ELBO		MCSA Variants			
				$N = 1$	$N = 10$	**pMCSA (ours)**	JSA	MSC	MSC-RB
yacht	403	6	277	**-2.45** ±0.01	**-2.44** ±0.01	**-2.49** ±0.01	-3.00 ±0.05	-2.98 ±0.04	-2.98 ±0.04
concrete	503	8	927	-3.25 ±0.01	**-3.24** ±0.01	**-3.20** ±0.01	-3.33 ±0.02	-3.32 ±0.02	-3.33 ±0.02
airfoil	353	6	1352	-2.53 ±0.02	-2.56 ±0.02	**-2.27** ±0.02	-2.51 ±0.02	-2.53 ±0.01	-2.51 ±0.01
energy	503	9	691	-2.42 ±0.02	-2.40 ±0.02	**-1.92** ±0.03	-2.38 ±0.02	-2.37 ±0.02	-2.36 ±0.02
wine	653	12	1439	**-0.96** ±0.01	**-0.96** ±0.01	**-0.95** ±0.01	**-0.97** ±0.01	**-0.97** ±0.01	**-0.97** ±0.01
boston	753	14	455	**-2.72** ±0.03	**-2.70** ±0.03	**-2.69** ±0.02	-2.82 ±0.02	-2.80 ±0.03	-2.78 ±0.02
sml	1203	23	3723	-1.32 ±0.01	**-1.25** ±0.02	**-1.22** ±0.01	-1.72 ±0.01	-1.97 ±0.02	-1.95 ±0.02
gas	6503	129	2308	**-0.06** ±0.01	**0.13** ±0.03	-0.09 ±0.02	-0.47 ±0.03	-0.47 ±0.04	-0.50 ±0.03

Table 4: Test Log Predictive Density on **Robust Gaussian Process Regression**

	D_{λ}	$D_{\mathbf{x}}$	N_{train}	ELBO	MCSA Variants			
				$N = 1$	**pMCSA (ours)**	JSA	MSC	MSC-RB
yacht	287	6	277	-3.63 ±0.02	**-3.31** ±0.04	**-3.29** ±0.05	**-3.25** ±0.04	**-3.27** ±0.05
airfoil	353	6	1352	-3.14 ±0.01	**-2.63** ±0.01	-2.83 ±0.04	-2.77 ±0.02	**-2.73** ±0.02
boston	472	13	455	**-2.98** ±0.01	**-2.96** ±0.02	**-3.00** ±0.03	**-3.00** ±0.03	**-2.96** ±0.03
energy	703	8	691	-2.75 ±0.01	**-2.58** ±0.03	-2.78 ±0.04	-2.70 ±0.04	-2.72 ±0.05
concrete	939	8	927	-3.68 ±0.01	**-3.49** ±0.01	-3.69 ±0.02	**-3.59** ±0.04	**-3.57** ±0.02
wine	1454	11	1439	-1.02 ±0.01	**-0.94** ±0.02	-1.04 ±0.01	-1.00 ±0.02	-0.99 ±0.02
gas	2440	128	2308	**0.18** ±0.02	**-0.86** ±0.02	-1.10 ±0.03	-1.10 ±0.04	-1.06 ±0.02

[1] D_{λ}: Dimentionality of $\boldsymbol{\lambda}$, $D_{\mathbf{x}}$: Number of features, N_{train}: Number of training data points.
[2] $\pm$ denotes the 95% bootstrap confidence intervals obtained from 20 replications.
[3] Bolded numbers don't have enough evidence to be distinguished from the best performing method under a .05 significance threshold (Friedman test with Nemenyi post-hoc test, Demšar 2006).

6.3 Bayesian Neural Network Regression

Setup For realistic experiments, we train Bayesian neural networks (BNN, Neal 1996) for regression. We use datasets from the UCI repository (Dua & Graff, 2017) with 90% random train-test splits. We run all methods with $T = 5 \cdot 10^4$ iterations. For the model, we use the priors and forward propagation method of Hernandez-Lobato & Adams (2015) with a 50-unit hidden layer (see Appendix C.1).

Results The results are shown in Table 3. pMCSA achieves the best performance compared to all other MCSA methods. Also, its overall performance is comparable to exclusive KL minimization methods (ELBO) unlike other MCSA methods. Furthermore, on airfoil and energy, pMCSA improves over ELBO by 0.29 `nat` and 0.48 `nat`. Even on gas where pMCSA did not beat ELBO, its performance is comparable, and it dominates all other MCSA methods by roughly 0.4 `nat`. Additional experimental results can be found in Appendix E.1

6.4 Robust Gaussian Process Regression

Setup We train Gaussian processes (GP) with a Student-T likelihood for robust regression. We datasets from the UCI repository (Dua & Graff, 2017) with 90% random train-test splits. We use the Matérn 5/2 covariance kernel with automatic relevance determination (Neal, 1996) (see Appendix C.2). We run all methods with $T = 2 \cdot 10^4$ iterations. For prediction, we use the mode of $q(\cdot; \boldsymbol{\lambda})$ for the hyperparameters and marginalize the latent function over $q(\cdot; \boldsymbol{\lambda})$ (Rasmussen & Williams, 2006). We consider ELBO with only $N = 1$ since differentiating through the likelihood makes its per-iteration cost comparable to MCSA methods with $N = 10$.

Results The results are shown in Table 4. Except for gas, pMCSA achieves better performance than all other methods. This suggests that, overall, the exclusive KL may be less effective for GP posteriors. Although ELBO achieves the best performance on gas, pMCSA dominates other MCSA methods. Our encouraging regression results suggest that incorporating methods such as inducing points (Snelson & Ghahramani, 2005) into MCSA may lead to an important new class of GP models. Additional experimental results can be found in Appendix E.2.

7 Related Works

Inclusive KL minimization Our MCSA framework generalizes MSC (Naesseth *et al.*, 2020) and JSA Ou & Song (2020), which are inclusive KL minimization based on SGD and Markov chains. Similar to MCSA is the method of Li *et al.* (2017). However, the convergence of this method is not guaranteed since it uses short Markov chains, disqualifying for MCSA. Other methods based on biased gradients have been proposed by Bornschein & Bengio (2015); Le *et al.* (2019), but these are specific for deep generative models. On a different note, Jerfel *et al.* (2021) use boosting instead of SGD to minimize the inclusive KL, which gradually builds a complex variational approximation from a simple variational family.

Beyond the KL Divergence Discovering alternative divergences for VI has been an active research area. For example, the χ^2 (Dieng *et al.*, 2017), f (Wan *et al.*, 2020; Wang *et al.*, 2018), α (Hernandez-Lobato *et al.*, 2016; Li & Turner, 2016; Regli & Silva, 2018), reguarlized importance ratio (Bamler *et al.*, 2017) divergences have been studied for VI. However, for gradient estimation these methods involve the importance ratio $w(\mathbf{z}) = \pi(\mathbf{z})/q(\mathbf{z})$, which leads to significant variance and low signal-to-noise ratio under model misspecification (Bamler *et al.*, 2017; Geffner & Domke, 2021a,c). In contrast, under stationarity, the variance of pMCSA is σ^2/N (σ is the variance of the score over the posterior) regardless of model misspecification. Meanwhile, Geffner & Domke (2021b); Ruiz & Titsias (2019); Salimans *et al.* (2015); Zhang *et al.* (2021) construct implicit divergences formed by MCMC. With the exception of Ruiz & Titsias (2019), most of these approaches aim to maximize auxiliary representations of the classic ELBO. Therefore, their property is likely to be similar to the exclusive KL.

Adaptive MCMC and MCSA As pointed out by Ou & Song (2020), using $q(\cdot;\boldsymbol{\lambda})$ within the MCMC kernel makes MCSA structurally equivalent to adaptive MCMC. In particular, Andrieu & Thoms (2008); Brofos *et al.* (2022); Gabrié *et al.* (2022); Garthwaite *et al.* (2016) discuss the use of stochastic approximation in adaptive MCMC. Also, Andrieu & Moulines (2006); Brofos *et al.* (2022); Giordani & Kohn (2010); Habib & Barber (2019); Holden *et al.* (2009); Keith *et al.* (2008); Neklyudov *et al.* (2019) specifically discuss adapting the propsosal of IMH kernels, and some of them use KL divergence minimization. These methods focus on showing ergodicity the samples ($\mathbf{z}^{(n)}$ in our notation) not the convergence of the variational approximation (or proposal in an adaptive MCMC context) $q(\cdot;\boldsymbol{\lambda})$. In this work, we focused on the convergence of $q(\cdot;\boldsymbol{\lambda})$, which could advance the adaptive MCMC side of the story.

8 Discussions

This paper presented a new theoretical framework for analyzing inclusive KL divergence minimization methods based on running SGD with Markov chains. Furthermore, we proposed pMCSA, a new MCSA method that enjoys substantially low variance. We have shown that this theoretical improvement translates into better empirical performance.

Limitations Our work has two main limitations. Firstly, since our work aims to understand existing MCSA methods, it inherits the current limitations of MCSA methods. For example, minibatch subsampling is challenging for models with non-factorizable likelihoods (Naesseth *et al.*, 2020). Secondly, our theoretical analysis in Section 4 requires Assumption 3, which is strong, but required to connect with MCGD. An important future direction would be to relax the assumptions needed by MCGD.

Towards Alternative Divergences In Section 6, we have shown that minimizing the *inclusive* KL is competitive against minimizing the *exclusive* KL on general Bayesian inference problems. Although Dhaka *et al.* (2021) has shown that the inclusive KL fails on high-dimensional problems, this is only the case under the presence of strong correlations. Before entirely ditching the inclusive KL, it is essential to ask, "how correlated posteriors really are in practice?" Furthermore, the true performance of alternative divergences is often masked by the limitations of the inference procedure (Geffner & Domke, 2021a,c). Given that pMCSA significantly advances the best-known performance of inclusive KL minimization, it is possible that similar improvements could be extracted from other divergences. To conclude, our results motivate further development of better inference algorithms for alternative divergence measures.

References

Andrieu, Christophe, & Moulines, Éric. 2006. On the Ergodicity Properties of Some Adaptive MCMC Algorithms. *The Annals of Applied Probability*, **16**(3).

Andrieu, Christophe, & Thoms, Johannes. 2008. A Tutorial on Adaptive MCMC. *Statistics and Computing*, **18**(4), 343–373.

Andrieu, Christophe, Lee, Anthony, & Vihola, Matti. 2018. Uniform Ergodicity of the Iterated Conditional SMC and Geometric Ergodicity of Particle Gibbs Samplers. *Bernoulli*, **24**(2).

Bamler, Robert, Zhang, Cheng, Opper, Manfred, & Mandt, Stephan. 2017. Perturbative Black Box Variational Inference. *In: Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc.

Blei, David M., Kucukelbir, Alp, & McAuliffe, Jon D. 2017. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, **112**(518), 859–877.

Bornschein, Jörg, & Bengio, Yoshua. 2015 (May). Reweighted Wake-Sleep. *In: Proceedings of the International Conference on Learning Representations*.

Bottou, Léon, Curtis, Frank E., & Nocedal, Jorge. 2018. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, **60**(2), 223–311.

Brofos, James, Gabrie, Marylou, Brubaker, Marcus A., & Lederman, Roy R. 2022. Adaptation of the Independent Metropolis-Hastings Sampler with Normalizing Flow Proposals. *Pages 5949–5986 of: Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, vol. 151. ML Research Press.

de Boer, Pieter-Tjerk, Kroese, Dirk P., Mannor, Shie, & Rubinstein, Reuven Y. 2005. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, **134**(1), 19–67.

Debavelaere, Vianney, Durrleman, Stanley, & Allassonnière, Stéphanie. 2021. On the Convergence of Stochastic Approximations under a Subgeometric Ergodic Markov Dynamic. *Electronic Journal of Statistics*, **15**(1).

Demšar, Janez. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, **7**(1), 1–30.

Dhaka, Akash Kumar, Catalina, Alejandro, Welandawe, Manushi, Andersen, Michael R., Huggins, Jonathan, & Vehtari, Aki. 2021. Challenges and Opportunities in High Dimensional Variational Inference. *In: Advances in Neural Information Processing Systems*.

Dieng, Adji Bousso, Tran, Dustin, Ranganath, Rajesh, Paisley, John, & Blei, David. 2017. Variational Inference via χ Upper Bound Minimization. *Pages 2729–2738 of: Advances in Neural Information Processing Systems*, vol. 30. Long Beach, California, USA: Curran Associates, Inc.

Doan, Thinh T., Nguyen, Lam M., Pham, Nhan H., & Romberg, Justin. 2020a (Oct.). *Convergence Rates of Accelerated Markov Gradient Descent with Applications in Reinforcement Learning*. Tech. rept. arXiv:2002.02873 [math]. ArXiv.

Doan, Thinh T., Nguyen, Lam M., Pham, Nhan H., & Romberg, Justin. 2020b (Apr.). *Finite-Time Analysis of Stochastic Gradient Descent under Markov Randomness*. Tech. rept. arXiv:2003.10973. ArXiv.

Domke, Justin, & Sheldon, Daniel R. 2018. Importance Weighting and Variational Inference. *In: Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc.

Dua, Dheeru, & Graff, Casey. 2017. UCI Machine Learning Repository.

Duchi, John C., Agarwal, Alekh, Johansson, Mikael, & Jordan, Michael I. 2012. Ergodic Mirror Descent. *SIAM Journal on Optimization*, **22**(4), 1549–1578.

Gabrié, Marylou, Rotskoff, Grant M., & Vanden-Eijnden, Eric. 2022. Adaptive Monte Carlo Augmented with Normalizing Flows. *Proceedings of the National Academy of Sciences*, **119**(10), e2109420119.

Garthwaite, P. H., Fan, Y., & 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, Hong, Xu, Kai, & Ghahramani, Zoubin. 2018. Turing: A Language for Flexible Probabilistic Inference. *Pages 1682–1690 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 84. ML Research Press.

Geffner, Tomas, & Domke, Justin. 2020. A Rule for Gradient Estimator Selection, with an Application to Variational Inference. *Pages 1803–1812 of: Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, vol. 108. ML Research Press.

Geffner, Tomas, & Domke, Justin. 2021a. Empirical Evaluation of Biased Methods for Alpha Divergence Minimization. *In: Proceedings of the Symposium on Advances in Approximate Bayesian Inference*.

Geffner, Tomas, & Domke, Justin. 2021b. MCMC Variational Inference via Uncorrected Hamiltonian Annealing. *Pages 639–651 of: Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc.

Geffner, Tomas, & Domke, Justin. 2021c. On the Difficulty of Unbiased Alpha Divergence Minimization. *Pages 3650–3659 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 139. ML Research Press.

Giordani, Paolo, & Kohn, Robert. 2010. Adaptive Independent Metropolis–Hastings by Fast Estimation of Mixtures of Normals. *Journal of Computational and Graphical Statistics*, **19**(2), 243–259.

Gu, Ming Gao, & Kong, Fan Hui. 1998. A Stochastic Approximation Algorithm with Markov Chain Monte-Carlo Method for Incomplete Data Estimation Problems. *Proceedings of the National Academy of Sciences*, **95**(13), 7270–7274.

Habib, Raza, & Barber, David. 2019. Auxiliary Variational MCMC. *In: Proceedings of the International Conference on Learning Representations*.

Hastings, W. K. 1970. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57**(1), 97–109.

Hernandez-Lobato, Jose, Li, Yingzhen, Rowland, Mark, Bui, Thang, Hernandez-Lobato, Daniel, & Turner, Richard. 2016. Black-Box Alpha Divergence Minimization. *Pages 1511–1520 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 48. New York, New York, USA: ML Research Press.

Hernandez-Lobato, Jose Miguel, & Adams, Ryan. 2015. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *Pages 1861–1869 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 37. Lille, France: ML Research Press.

Hesterberg, Tim. 1995. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics*, **37**(2), 185–194.

Holden, Lars, Hauge, Ragnar, & Holden, Marit. 2009. Adaptive Independent Metropolis–Hastings. *The Annals of Applied Probability*, **19**(1).

Jerfel, Ghassen, Wang, Serena, Wong-Fannjiang, Clara, Heller, Katherine A., Ma, Yian, & Jordan, Michael I. 2021. Variational Refinement for Importance Sampling Using the Forward Kullback-Leibler Divergence. *Pages 1819–1829 of: Proceedings of the International Conference on Uncertainty in Artifical Intelligence*. PMLR, vol. 161. ML Research Press.

Karimi, Belhal, Miasojedow, Blazej, Moulines, Eric, & Wai, Hoi-To. 2019. Non-Asymptotic Analysis of Biased Stochastic Approximation Scheme. *Pages 1944–1974 of: Proceedings of the Annual Conference on Learning Theory*. PMLR, vol. 99. ML Research Press.

Keith, Jonathan M., Kroese, Dirk P., & Sofronov, George Y. 2008. Adaptive Independence Samplers. *Statistics and Computing*, **18**(4), 409–420.

Kingma, Diederik P., & Ba, Jimmy. 2015. Adam: A Method for Stochastic Optimization. *In: Proceedings of the International Conference on Learning Representations*.

Kucukelbir, Alp, Tran, Dustin, Ranganath, Rajesh, Gelman, Andrew, & Blei, David M. 2017. Automatic Differentiation Variational Inference. *Journal of Machine Learning Research*, **18**(14), 1–45.

Le, Tuan Anh, Kosiorek, Adam R., Siddharth, N., Teh, Yee Whye, & Wood, Frank. 2019 (July). Revisiting Reweighted Wake-Sleep for Models with Stochastic Control Flow. *In: Proceedings of the International Conference on Uncertainty in Artifical Intelligence*.

Li, Yingzhen, & Turner, Richard E. 2016. Rényi Divergence Variational Inference. *In: Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc.

Li, Yingzhen, Turner, Richard E., & Liu, Qiang. 2017 (May). *Approximate Inference with Amortised MCMC*. Tech. rept. arXiv:1702.08343 [cs, stat]. ArXiv.

MacKay, David J.C. 2001 (June). *Local Minima, Symmetry-Breaking, and Model Pruning in Variational Free Energy Minimization*. Technical Report.

Mengersen, K. L., & Tweedie, R. L. 1996. Rates of Convergence of the Hastings and Metropolis Algorithms. *The Annals of Statistics*, **24**(1), 101–121.

Minka, Tom. 2005 (Jan.). *Divergence Measures and Message Passing*. Tech. rept. MSR-TR-2005-173. Microsoft Research.

Naesseth, Christian, Lindsten, Fredrik, & Blei, David. 2020. Markovian Score Climbing: Variational Inference with KL(p||q). *Pages 15499–15510 of: Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc.

Neal, Radford M. 1996. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics, vol. 118. New York, NY: Springer New York.

Neklyudov, Kirill, Egorov, Evgenii, Shvechikov, Pavel, & Vetrov, Dmitry. 2019 (June). *Metropolis-Hastings View on Variational Inference and Adversarial Training*. Tech. rept. arXiv:1810.07151 [cs, stat]. ArXiv.

Nesterov, Yurii Evgen'evich. 1983. A Method of Solving a Convex Programming Problem with Convergence Rate {$O(\frac{1}{k^2})$}. *Doklady Akademii Nauk SSSR*, **269**(3), 543–547.

Nguyen, Lam, Nguyen, Phuong Ha, van Dijk, Marten, Richtarik, Peter, Scheinberg, Katya, & Takac, Martin. 2018. SGD and Hogwild! Convergence without the Bounded Gradients Assumption. *Pages 3750–3758 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 80. ML Research Press.

Ou, Zhijian, & Song, Yunfu. 2020. Joint Stochastic Approximation and Its Application to Learning Discrete Latent Variable Models. *Pages 929–938 of: Proceedings of the International Conference on Uncertainty in Artifical Intelligence*. PMLR, vol. 124. ML Research Press.

Polyak, B.T. 1964. Some Methods of Speeding up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, **4**(5), 1–17.

Ram, S. Sundhar, Nedić, A., & Veeravalli, V. V. 2009. Incremental Stochastic Subgradient Algorithms for Convex Optimization. *SIAM Journal on Optimization*, **20**(2), 691–717.

Ranganath, Rajesh, Gerrish, Sean, & Blei, David. 2014. Black Box Variational Inference. *Pages 814–822 of: Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, vol. 33. Reykjavik, Iceland: ML Research Press.

Rasmussen, Carl Edward, & Williams, Christopher K. I. 2006. *Gaussian Processes for Machine Learning*. Adaptive Comput. Mach. Learn. Cambridge, Mass: MIT Press.

Regli, Jean-Baptiste, & Silva, Ricardo. 2018. Alpha-Beta Divergence for Variational Inference. May.

Robbins, Herbert, & Monro, Sutton. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, **22**(3), 400–407.

Robert, Christian P., & Casella, George. 2004. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. New York, NY: Springer New York.

Roeder, Geoffrey, Wu, Yuhuai, & Duvenaud, David K. 2017. Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference. *In: Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc.

Ruiz, Francisco, & Titsias, Michalis. 2019. A Contrastive Divergence for Combining Variational Inference and MCMC. *Pages 5537–5545 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 97. ML Research Press.

Salimans, Tim, Kingma, Diederik, & Welling, Max. 2015. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. *Pages 1218–1226 of: Proceedings of the International Conference on Machine Learning*. PMLR, vol. 37. Lille, France: ML Research Press.

Smith, Richard L., & Tierney, Luke. 1996. *Exact Transition Probabilities for the Independence Metropolis Sampler*. Tech. rept.

Snelson, Edward, & Ghahramani, Zoubin. 2005. Sparse Gaussian Processes Using Pseudo-Inputs. *In: Advances in Neural Information Processing Systems*, vol. 18. MIT Press.

Sun, Tao, Sun, Yuejiao, & Yin, Wotao. 2018. On Markov Chain Gradient Descent. *In: Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc.

Tadić, Vladislav B., & Doucet, Arnaud. 2017. Asymptotic Bias of Stochastic Gradient Search. *The Annals of Applied Probability*, **27**(6).

Trippe, Brian, & Turner, Richard. 2017. *Overpruning in Variational Bayesian Neural Networks*. Tech. rept. arXiv:1801.06230. ArXiv.

Wan, Neng, Li, Dapeng, & Hovakimyan, Naira. 2020. F-Divergence Variational Inference. *Pages 17370–17379 of: Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc.

Wang, Dilin, Liu, Hao, & Liu, Qiang. 2018. Variational Inference with Tail-Adaptive f-Divergence. *In: Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc.

Wang, Guanyang. 2020. Exact Convergence Rate Analysis of the Independent Metropolis-Hastings Algorithms. *To appear in Bernoulli*, Dec.

Xiong, Huaqing, Xu, Tengyu, Liang, Yingbin, & Zhang, Wei. 2021. Non-Asymptotic Convergence of Adam-Type Reinforcement Learning Algorithms under Markovian Sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(12), 10460–10468.

Zhang, Guodong, Hsu, Kyle, Li, Jianing, Finn, Chelsea, & Grosse, Roger B. 2021. Differentiable Annealed Importance Sampling and the Perils of Gradient Noise. *Pages 19398–19410 of: Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   See the contribution summary in Section 1.
   (b) Did you describe the limitations of your work? [Yes]
   See Section 8.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   The key assumptions are stated in each proof.
   (b) Did you include complete proofs of all theoretical results? [Yes]
   See Appendix D.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   It is included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   See Section 6. Additional details can be found in the code in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   See the text in Section 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
   See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [No]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Computational Resources

Table 5: Computational Resources for Bayesian Neural Network Regression

Type	Model and Specifications
System Topology	4 nodes with 20 logical threads each
Processor	Intel Xeon Xeon E5–2640 v4, 2.2 GHz (maximum 3.1 GHz)
Cache	32 kB L1, 256 kB L2, and 25 MB L3
Memory	64GB RAM

Table 6: Computational Resources for Robust Gaussian Process Regression

Type	Model and Specifications
System Topology	1 node with 16 logical threads
Processor	AMD EPYC 7262, 3.2 GHz (maximum 3.4 GHz)
Accelerator	NVIDIA Titan RTX, 1.3 GHZ, 24GB RAM
Cache	256 kB L1, 4MiB L2, and 128MiB L3
Memory	126GB RAM

B Pseudocodes

B.1 Markov Chain Kernels

Algorithm 1: Conditional Importance Sampling Kernel

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

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

$\mathbf{z}^{(i)} \sim q_{\text{def.}}(\mathbf{z}; \boldsymbol{\lambda}_{t-1}) \quad \text{for } i = 1, 2, \dots, N$

$\widetilde{w}(\mathbf{z}^{(i)}) = p(\mathbf{z}^{(i)}, \mathbf{x}) \,/\, q_{\text{def.}}\left(\mathbf{z}^{(i)}; \boldsymbol{\lambda}_{t-1}\right) \quad \text{for } i = 0, 1, \dots, N$

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

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

Algorithm 2: Independent Metropolis-Hastings Kernel

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

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

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

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

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

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

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

else

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

end

B.2 Markov Chain Score Ascent Algorithms

Algorithm 3: Markovian Score Climbing

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**
  $\mathbf{z}_t \sim K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_{t-1}, \cdot)$
  $\mathbf{g}(\boldsymbol{\lambda}) = \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_t)$
  $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, \mathbf{g}(\boldsymbol{\lambda}_{t-1})$
end

Algorithm 4: Joint Stochastic Approximation

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

for $t = 1, 2, \ldots, T$ **do**
  $\mathbf{z}_t^{(0)} = \mathbf{z}_{t-1}^{(N)}$
  for $n = 1, 2, \ldots, N$ **do**
    $\mathbf{z}_t^{(n)} \sim K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_t^{(n-1)}, \cdot)$
  end
  $\mathbf{g}(\boldsymbol{\lambda}) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_t^{(n)})$
  $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, \mathbf{g}(\boldsymbol{\lambda}_{t-1})$
end

Algorithm 5: Parallel Markov Chain Score Ascent

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 $n = 1, 2, \ldots, N$ **do**
    $\mathbf{z}_t^{(n)} \sim K_{\boldsymbol{\lambda}_{t-1}}(\mathbf{z}_{t-1}^{(n)}, \cdot)$
  end
  $\mathbf{g}(\boldsymbol{\lambda}) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_t^{(n)})$
  $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + \gamma_t \, \mathbf{g}(\boldsymbol{\lambda}_{t-1})$
end

C Probabilistic Models Used in the Experiments

C.1 Bayesian Neural Network Regression

We use the BNN model of Hernandez-Lobato & Adams (2015) defined as

$$
\begin{aligned}
\lambda^{-1} &\sim \text{inverse-gamma}\,(\alpha = 6, \beta = 6) \\
\gamma^{-1} &\sim \text{inverse-gamma}\,(\alpha = 6, \beta = 6) \\
\mathbf{W}_1 &\sim \mathcal{N}\left(\mathbf{0}, \lambda^{-1}\mathbf{I}\right) \\
\mathbf{z} &= \text{ReLU}\,(\mathbf{W}_1 \mathbf{x}_i) \\
\mathbf{W}_2 &\sim \mathcal{N}\left(\mathbf{0}, \lambda^{-1}\mathbf{I}\right) \\
\hat{y} &= \text{ReLU}\,(\mathbf{W}_2 \mathbf{z}) \\
y_i &\sim \mathcal{N}\left(\hat{y}, \gamma^{-1}\right)
\end{aligned}
$$

where $\mathbf{x}_i$ and y_i are the feature vector and target value of the ith datapoint. Given the variational distribution of $\lambda^{-1}, \gamma^{-1}, \mathbf{W}_1, \mathbf{W}_2$, we use the same posterior predictive approximation of Hernandez-Lobato & Adams (2015). We apply z-standardization (whitening) to the features $\mathbf{x}_i$ and the target values y_i, and unwhiten the predictive distribution.

C.2 Robust Gaussian Process Logistic Regression

We perform robust Gaussian process regression by using a student-t prior with a latent Gaussian process prior. The model is defined as

$$\begin{aligned}
\log \sigma_f &\sim \mathcal{N}(0, 4) \\
\log \epsilon &\sim \mathcal{N}(0, 4) \\
\log \ell_i &\sim \mathcal{N}(0, 0.2) \\
f &\sim \mathcal{GP}\left(\mathbf{0}, \mathbf{\Sigma}_{\sigma_f, \ell} + \left(\delta + \epsilon^2\right) \mathbf{I}\right) \\
\nu &\sim \mathsf{gamma}\left(\alpha = 4, \beta = 1/10\right) \\
\log \sigma_y &\sim \mathcal{N}(0, 4) \\
y_i &\sim \mathsf{student\text{-}t}\left(f\left(\mathbf{x}_i\right), \sigma_y, \nu\right).
\end{aligned}$$

The covariance $\mathbf{\Sigma}$ is computed using a kernel $k(\cdot, \cdot)$ such that $[\mathbf{\Sigma}]_{i,j} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$ 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

$$k\left(\mathbf{x}, \mathbf{x}';\ \sigma^2, \ell_1^2, \ldots, \ell_D^2\right) = \sigma_f \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right) \quad \text{where} \quad r = \sum_{i=1}^{D} \frac{\left(\mathbf{x}_i - \mathbf{x}_i'\right)^2}{\ell_i^2}$$

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

D Proofs

Proposition 1. Let $\boldsymbol{\eta} = \left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N)}\right)$ and a Markov chain kernel $P_{\boldsymbol{\lambda}}\left(\boldsymbol{\eta}, \cdot\right)$ be Π-invariant where Π is defined as

$$\Pi\left(\boldsymbol{\eta}\right) = \pi\left(\mathbf{z}^{(1)}\right)\pi\left(\mathbf{z}^{(2)}\right) \times \ldots \times \pi\left(\mathbf{z}^{(N)}\right).$$

Then, by defining the target function f and the gradient estimator $\mathbf{g}$ to be

$$f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right) = -\frac{1}{N}\sum_{n=1}^{N} \log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) - \mathbb{H}\left[\,\pi\,\right], \quad \text{and} \quad \mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right) = -\frac{1}{N}\sum_{n=1}^{N} \mathbf{s}\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)$$

where $\mathbb{H}\left[\,\pi\,\right]$ is the entropy of π, MCGD results in inclusive KL minimization as

$$\mathbb{E}_{\Pi}\left[\,f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right] = d_{\mathrm{KL}}(\pi \parallel q\left(\cdot; \boldsymbol{\lambda}\right)), \quad \text{and} \quad \mathbb{E}_{\Pi}\left[\,\mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right] = \nabla_{\boldsymbol{\lambda}} d_{\mathrm{KL}}(\pi \parallel q\left(\cdot; \boldsymbol{\lambda}\right)).$$

Proof of Proposition 1. For notational convenience, we define the shorthand

$$\pi\left(\mathbf{z}^{(1:N)}\right) = \pi\left(\mathbf{z}^{(1)}\right)\pi\left(\mathbf{z}^{(2)}\right) \times \ldots \times \pi\left(\mathbf{z}^{(N)}\right).$$

Then,

$$\begin{aligned}
&\mathbb{E}_{\Pi}\left[\,f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right] \\
&= \int \left(-\frac{1}{N}\sum_{n=1}^{N} \log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) - \mathbb{H}\left[\,\pi\,\right]\right) \pi\left(\mathbf{z}^{(1:N)}\right) d\mathbf{z}^{(1:N)} \\
&= \int \left(-\frac{1}{N}\sum_{n=1}^{N} \log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)\right) \pi\left(\mathbf{z}^{(1:N)}\right) d\mathbf{z}^{(1:N)} - \mathbb{H}\left[\,\pi\,\right] && \textit{Pulled out constant} \\
&= \frac{1}{N}\sum_{n=1}^{N} \left\{\int \left(-\log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)\right) \pi\left(\mathbf{z}^{(1:N)}\right) d\mathbf{z}^{(1:N)}\right\} - \mathbb{H}\left[\,\pi\,\right] && \textit{Swapped integral and sum} \\
&= \frac{1}{N}\sum_{n=1}^{N} \int \left(-\log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)\right) \pi\left(\mathbf{z}^{(n)}\right) d\mathbf{z}^{(n)} - \mathbb{H}\left[\,\pi\,\right] && \textit{Marginalized } \mathbf{z}^{(m)} \textit{ for all } m \neq n \\
&= \frac{1}{N}\sum_{n=1}^{N} \int \left(-\log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) + \log \pi\left(\mathbf{z}^{(n)}\right)\right) \pi\left(\mathbf{z}^{(n)}\right) d\mathbf{z}^{(n)} && \textit{Definition of entropy } \mathbb{H}\left[\pi\right] \\
&= \frac{1}{N}\sum_{n=1}^{N} \int \pi\left(\mathbf{z}^{(n)}\right) \log \frac{\pi\left(\mathbf{z}^{(n)}\right)}{q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right)} d\mathbf{z}^{(n)} \\
&= \frac{1}{N}\sum_{n=1}^{N} d_{\mathrm{KL}}(\pi \parallel q\left(\cdot; \boldsymbol{\lambda}\right)) && \textit{Definition of } d_{KL} \\
&= d_{\mathrm{KL}}(\pi \parallel q\left(\cdot; \boldsymbol{\lambda}\right)) && (4)
\end{aligned}$$

For $\mathbb{E}_{\Pi}\left[\,\mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right]$, note that

$$\nabla_{\boldsymbol{\lambda}} f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right) = -\frac{1}{N}\sum_{n=1}^{N} \nabla_{\boldsymbol{\lambda}} \log q\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) = -\frac{1}{N}\sum_{n=1}^{N} \mathbf{s}\left(\mathbf{z}^{(n)}; \boldsymbol{\lambda}\right) = \mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right). \tag{5}$$

Therefore, it suffices to show that

$$\begin{aligned}
\nabla_{\boldsymbol{\lambda}} d_{\mathrm{KL}}(\pi \parallel q\left(\cdot; \boldsymbol{\lambda}\right)) &= \nabla_{\boldsymbol{\lambda}} \mathbb{E}_{\Pi}\left[\,f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right] && \textit{Equation } (4) \\
&= \mathbb{E}_{\Pi}\left[\,\nabla_{\boldsymbol{\lambda}} f\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right] && \textit{Leibniz derivative rule} \\
&= \mathbb{E}_{\Pi}\left[\,\mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\,\right]. && \textit{Equation } (5)
\end{aligned}$$

□

Proposition 2. The maximum importance weight $w^* = \sup_{\mathbf{z}} w(\mathbf{z}) = \sup_{\mathbf{z}} \pi(\mathbf{z}) / q(\mathbf{z}; \boldsymbol{\lambda})$ is bounded below exponentially by the KL divergence as

$$\exp\left(d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda}))\right) < w^*.$$

Proof of Proposition 2.

$$\begin{aligned}
d_{\mathrm{KL}}(\pi \parallel q(\cdot; \boldsymbol{\lambda})) &= \mathbb{E}_{\mathbf{z} \sim \pi(\cdot)}\left[\log \frac{\pi(\mathbf{z})}{q(\mathbf{z}; \boldsymbol{\lambda})}\right] && \textit{Definition of } d_{KL} \\
&\leq \log \mathbb{E}_{\mathbf{z} \sim \pi(\cdot)}\left[\frac{\pi(\mathbf{z})}{q(\mathbf{z}; \boldsymbol{\lambda})}\right] && \textit{Jensen's inequality} \\
&\leq \log \mathbb{E}_{\mathbf{z} \sim \pi(\cdot)}\left[\, w^* \,\right] \\
&= \log w^*.
\end{aligned}$$

□

Lemma 1. *For the probability measures $p_1, \ldots, p_N$ and $q_1, \ldots, q_N$ defined on a measurable space $(\mathsf{X}, \mathcal{A})$ and an arbitrary set $A \in \mathcal{A}$,*

$$\begin{aligned}
&\left| \int_{A^N} p_1(dx_1)\, p_2(dx_2) \times \ldots \times p_N(dx_N) - q_1(dx_1)\, q_2(dx_2) \times \ldots \times q_N(dx_N) \right| \\
&\quad \leq \sum_{n=1}^{N} \left| \int_A p_n(dx_n) - q_n(dx_n) \right|
\end{aligned}$$

*Proof of **Lemma** 1.* By using the following shorthand notations

$$\begin{aligned}
p_{(1:N)}\left(dx_{(1:N)}\right) &= p_1(dx_1)\, p_2(dx_2) \times \ldots \times p_N(dx_N) \\
q_{(1:N)}\left(dx_{(1:N)}\right) &= q_1(dx_1)\, q_2(dx_2) \times \ldots \times q_N(dx_N),
\end{aligned}$$

the result follows from induction as

$$\begin{aligned}
&\left| \int_{A^N} p_{(1:N)}\left(dx_{(1:N)}\right) - q_{(1:N)}\left(dx_{(1:N)}\right) \right| \\
&\quad = \Bigg| \left(\int_A p_1(dx_1) - q_1(dx_1) \right) \int_{A^{N-1}} p_{(2:N)}\left(dx_{(2:N)}\right) \\
&\qquad + \int_A q_1(dx_1) \left(\int_{A^{N-1}} p_{(2:N)}\left(dx_{(2:N)}\right) - q_{(2:N)}\left(dx_{(2:N)}\right) \right) \Bigg| \\
&\quad \leq \left| \int_A p_1(dx_1) - q_1(dx_1) \right| \int_{A^{N-1}} p_{(2:N)}\left(dx_{(2:N)}\right) \\
&\qquad + \int_A q_1(dx_1) \left| \int_{A^{N-1}} p_{(2:N)}\left(dx_{(2:N)}\right) - q_{(2:N)}\left(dx_{(2:N)}\right) \right| && \textit{Triangle inequality} \\
&\quad \leq \left| \int_A p_1(dx_1) - q_1(dx_1) \right| \\
&\qquad + \left| \int_{A^{N-1}} p_{(2:N)}\left(dx_{(2:N)}\right) - q_{(2:N)}\left(dx_{(2:N)}\right) \right|. && \textit{Applied } p_n(A), q_n(A) \leq 1
\end{aligned}$$

□

Theorem 1. MSC (Naesseth *et al.*, 2020) is obtained by defining

$$P_\lambda^k(\boldsymbol{\eta}, d\boldsymbol{\eta}') = K_\lambda^k(\mathbf{z}, d\mathbf{z}')$$

with $\boldsymbol{\eta}_t = \mathbf{z}_t$ where $K_{\boldsymbol{\lambda}}(\mathbf{z}, \cdot)$ is the CIS kernel with $q_{def.}(\cdot; \boldsymbol{\lambda})$ as its proposal distribution. Then, given Assumption 2 and 3, the mixing rate and the gradient bounds are given as

$$d_{\mathrm{TV}}\left(P_\lambda^k(\boldsymbol{\eta}, \cdot), \Pi\right) \leq \left(1 - \frac{N-1}{2w^* + N - 2}\right)^k \quad \text{and} \quad \mathbb{E}\left[\, \| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \,\middle|\, \mathcal{F}_t \,\right] \leq L^2,$$

where $w^* = \sup_{\mathbf{z}} \pi(\mathbf{z}) / q_{\mathrm{def.}}(\mathbf{z}; \boldsymbol{\lambda})$.

Proof of Theorem 1. MSC is described in Algorithm 3. At each iteration, it performs a single MCMC transition with the CIS kernel where it internally uses N proposals.

Ergodicity of the Markov Chain The ergodic convergence rate of $P_{\boldsymbol{\lambda}}$ is equal to that of $K_{\boldsymbol{\lambda}}$, the CIS kernel proposed by Naesseth *et al.* (2020). Although not mentioned by Naesseth *et al.* (2020), this kernel has been previously proposed as the iterated sequential importance resampling (i-SIR) by Andrieu *et al.* (2018) with its corresponding geometric convergence rate as

$$d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta},\cdot\right),\Pi\right)=d_{\mathrm{TV}}\left(K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z},\cdot\right),\pi\right)\leq\left(1-\frac{N-1}{2w^{*}+N-2}\right)^{k}.$$

Bound on the Gradient Variance The bound on the gradient variance is straightforward given Assumption 3. For simplicity, we denote the rejection state as $\mathbf{z}^{(1)}=\mathbf{z}_{t-1}$. Then,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{g}\left(\boldsymbol{\lambda},\boldsymbol{\eta}\right)\right\|_{*}^{2}\middle|\mathcal{F}_{t}\right]\\
&=\mathbb{E}\left[\left\|\mathbf{g}\left(\boldsymbol{\lambda},\boldsymbol{\eta}\right)\right\|_{*}^{2}\middle|\mathcal{F}_{t}\right]\\
&=\mathbb{E}_{\mathbf{z}\sim K_{\boldsymbol{\lambda}_{t-1}}\left(\mathbf{z}_{t-1},\cdot\right)}\left[\left\|\mathbf{s}\left(\boldsymbol{\lambda};\mathbf{z}\right)\right\|_{*}^{2}\middle|\boldsymbol{\lambda}_{t-1},\mathbf{z}_{t-1}\right]\\
&=\int\sum_{n=1}^{N}\frac{w\left(\mathbf{z}^{(n)}\right)}{\sum_{m=1}^{N}w\left(\mathbf{z}^{(m)}\right)}\left\|\mathbf{s}\left(\cdot;\mathbf{z}^{(n)}\right)\right\|_{*}^{2}\prod_{n=2}^{N}q\left(d\mathbf{z}^{(n)};\boldsymbol{\lambda}_{t-1}\right) && \textit{(Andrieu et al., 2018)}\\
&\leq L^{2}\int\sum_{n=1}^{N}\frac{w\left(\mathbf{z}^{(n)}\right)}{\sum_{m=1}^{N}w\left(\mathbf{z}^{(m)}\right)}\prod_{n=2}^{N}q\left(d\mathbf{z}^{(n)};\boldsymbol{\lambda}_{t-1}\right) && \textit{Assumption 3}\\
&=L^{2}\int\prod_{n=2}^{N}q\left(d\mathbf{z}^{(n)};\boldsymbol{\lambda}_{t-1}\right) && \textit{The sum of weights is 1}\\
&=L^{2}.
\end{aligned}$$

□

Lemma 2. *For $w^{*}=\sup_{\mathbf{z}}w\left(\mathbf{z}\right)$, $\lambda\left(\cdot\right)$ in Equation (3) is bounded as*

$$\max\left(1-\frac{1}{w},0\right)\leq\lambda\left(w\right)\leq1-\frac{1}{w^{*}}.$$

*Proof of **Lemma** 2.* The proof can be found in the proof of Theorem 3 of Smith & Tierney (1996). □

Lemma 3. *For $w^{*}=\sup_{\mathbf{z}}w\left(\mathbf{z}\right)$, $T_{n}\left(\cdot\right)$ in Equation (3) is bounded as*

$$T_{n}\left(w\right)\leq\frac{n}{w}\left(1-\frac{1}{w^{*}}\right)^{n-1}.$$

*Proof of **Lemma** 3.*

$$\begin{aligned}
T_{n}\left(w\right)&=\int_{w}^{\infty}\frac{n}{v^{2}}\lambda^{n-1}\left(v\right)\,dv && \textit{Equation (3)}\\
&\leq\int_{w}^{\infty}\frac{n}{v^{2}}\left(1-\frac{1}{w^{*}}\right)^{n-1}dv && \textit{Lemma 2}\\
&=n\left(1-\frac{1}{w^{*}}\right)^{n-1}\int_{w}^{\infty}\frac{1}{v^{2}}\,dv && \textit{Pulled out constant}\\
&=n\left(1-\frac{1}{w^{*}}\right)^{n-1}\left(-\frac{1}{v}\Big|_{w}^{\infty}\right) && \textit{Solved indefinite integral}
\end{aligned}$$

$$= \frac{n}{w}\left(1-\frac{1}{w^*}\right)^{n-1}.$$

This upper bound is in general difficult to improve unless we impose stronger assumptions on π and q. □

Lemma 4. *For a positive test function $f : \mathcal{Z} \to \mathbb{R}^+$, the estimate of a π-invariant independent Metropolis-Hastings kernel with a proposal q is bounded as*

$$\mathbb{E}_{K^n(\mathbf{z},\cdot)}\left[\, f \,|\, \mathbf{z} \,\right] \leq n\left(1-\frac{1}{w^*}\right)^{n-1} \mathbb{E}_q\left[\, f \,\right] + \left(1-\frac{1}{w^*}\right)^{n} f\left(\mathbf{z}\right)$$

where $w\left(\mathbf{z}\right) = \pi\left(\mathbf{z}\right)/q\left(\mathbf{z}\right)$ and $w^* = \sup_{\mathbf{z}} w\left(\mathbf{z}\right)$.

*Proof of **Lemma** 4.*

$$\begin{aligned}
&\mathbb{E}_{K^n(\mathbf{z},\cdot)}\left[\, f \,|\, \mathbf{z} \,\right] \\
&= \int T_n\left(w\left(\mathbf{z}\right) \vee w\left(\mathbf{z}'\right)\right) f\left(\mathbf{z}'\right) \pi\left(\mathbf{z}'\right) d\mathbf{z}' + \lambda^n\left(w\left(\mathbf{z}\right)\right) f\left(\mathbf{z}\right) && \textit{Equation (2)} \\
&\leq \int \frac{n}{w\left(\mathbf{z}\right) \vee w\left(\mathbf{z}'\right)}\left(1-\frac{1}{w^*}\right)^{n-1} f\left(\mathbf{z}'\right) \pi\left(\mathbf{z}'\right) d\mathbf{z}' + \lambda^n\left(w\left(\mathbf{z}\right)\right) f\left(\mathbf{z}\right) && \textit{Lemma 3} \\
&\leq \int \frac{n}{w\left(\mathbf{z}'\right)}\left(1-\frac{1}{w^*}\right)^{n-1} f\left(\mathbf{z}'\right) \pi\left(\mathbf{z}'\right) d\mathbf{z}' + \lambda^n\left(w\left(\mathbf{z}\right)\right) f\left(\mathbf{z}\right) && \frac{1}{w\left(\mathbf{z}\right) \vee w\left(\mathbf{z}'\right)} \leq \frac{1}{w\left(\mathbf{z}'\right)} \\
&= n\left(1-\frac{1}{w^*}\right)^{n-1} \int \frac{1}{w\left(\mathbf{z}'\right)} f\left(\mathbf{z}'\right) \pi\left(\mathbf{z}'\right) d\mathbf{z}' + \lambda^n\left(w\left(\mathbf{z}\right)\right) f\left(\mathbf{z}\right) && \textit{Pulled out constant} \\
&= n\left(1-\frac{1}{w^*}\right)^{n-1} \int f\left(\mathbf{z}'\right) q\left(\mathbf{z}'\right) d\mathbf{z}' + \lambda^n\left(w\left(\mathbf{z}\right)\right) f\left(\mathbf{z}\right) && \textit{Definition of } w\left(\mathbf{z}\right) \\
&\leq n\left(1-\frac{1}{w^*}\right)^{n-1} \int f\left(\mathbf{z}'\right) q\left(\mathbf{z}'\right) d\mathbf{z}' + \left(1-\frac{1}{w^*}\right)^{n} f\left(\mathbf{z}\right) && \textit{Lemma 2} \\
&= n\left(1-\frac{1}{w^*}\right)^{n-1} \mathbb{E}_q\left[\, f \,\right] + \left(1-\frac{1}{w^*}\right)^{n} f\left(\mathbf{z}\right).
\end{aligned}$$

□

Theorem 2. JSA (Ou & Song, 2020) is obtained by defining

$$P_{\boldsymbol{\lambda}}^k\left(\boldsymbol{\eta}, d\boldsymbol{\eta}'\right) = K_{\boldsymbol{\lambda}}^{N(k-1)+1}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{(1)}\right) K_{\boldsymbol{\lambda}}^{N(k-1)+2}\left(\mathbf{z}^{(2)}, d\mathbf{z}'^{(2)}\right) \cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{N(k-1)+N}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{(N)}\right)$$

with $\boldsymbol{\eta}_t = \left(\mathbf{z}_t^{(1)}, \mathbf{z}_t^{(2)}, \ldots, \mathbf{z}_t^{(N)}\right)$. Then, given Assumption 2 and 3, the mixing rate and the gradient variance bounds are

$$d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^k\left(\boldsymbol{\eta}, \cdot\right), \Pi\right) \leq C\left(\rho, N\right) \rho^{kN} \quad \text{and} \quad \mathbb{E}\left[\, \left\| \mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right) \right\|_*^2 \,\middle|\, \mathcal{F}_t \,\right] \leq L^2 \left[\, \frac{1}{2} + \frac{3}{2}\frac{1}{N} + \mathcal{O}\left(1/w^*\right) \,\right],$$

where $w^* = \sup_{\mathbf{z}} \pi\left(\mathbf{z}\right)/q_{\text{def.}}\left(\mathbf{z}; \boldsymbol{\lambda}\right)$ and $C\left(\rho, N\right) > 0$ is a finite constant depending on ρ and N.

Proof of Theorem 2. JSA is described in Algorithm 4. At each iteration, it performs N MCMC transitions, and uses the N samples to estimate the gradient.

Ergodicity of the Markov Chain The state transitions of the Markov chain samples $\mathbf{z}^{(1:N)}$ are visualized as

	$\mathbf{z}_t^{(1)}$	$\mathbf{z}_t^{(2)}$	$\mathbf{z}_t^{(3)}$	$\ldots$	$\mathbf{z}_t^{(N)}$
$t=1$	$K_{\lambda_1}\left(\mathbf{z}_0, d\mathbf{z}_1^{(1)}\right)$	$K_{\lambda_1}^2\left(\mathbf{z}_0, d\mathbf{z}_1^{(2)}\right)$	$K_{\lambda_1}^3\left(\mathbf{z}_0, d\mathbf{z}_1^{(3)}\right)$	$\ldots$	$K_{\lambda_1}^N\left(\mathbf{z}_0, d\mathbf{z}_1^{(N)}\right)$
$t=2$	$K_{\lambda_2}^{N+1}\left(\mathbf{z}_0, d\mathbf{z}_2^{(1)}\right)$	$K_{\lambda_2}^{N+2}\left(\mathbf{z}_0, d\mathbf{z}_2^{(2)}\right)$	$K_{\lambda_2}^{N+3}\left(\mathbf{z}_0, d\mathbf{z}_2^{(3)}\right)$	$\ldots$	$K_{\lambda_2}^{2N}\left(\mathbf{z}_0, d\mathbf{z}_2^{(N)}\right)$
$\vdots$			$\vdots$		
$t=k$	$K_{\lambda_k}^{(k-1)N+1}\left(\mathbf{z}_0, d\mathbf{z}_k^{(1)}\right)$	$K_{\lambda_k}^{(k-1)N+2}\left(\mathbf{z}_0, d\mathbf{z}_k^{(2)}\right)$	$K_{\lambda_k}^{(k-1)N+3}\left(\mathbf{z}_0, d\mathbf{z}_k^{(3)}\right)$	$\ldots$	$K_{\lambda_k}^{(k-1)N+N}\left(\mathbf{z}_0, d\mathbf{z}_k^{(N)}\right)$

where $K_{\boldsymbol{\lambda}}(\mathbf{z}, \cdot)$ is an IMH kernel. Therefore, the n-step transition kernel for the vector of the Markov-chain samples $\boldsymbol{\eta} = \mathbf{z}^{(1:N)}$ is represented as

$$P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, d\boldsymbol{\eta}'\right) = K_{\boldsymbol{\lambda}}^{N(k-1)+1}\left(\mathbf{z}_1, d\mathbf{z}_1'\right) K_{\boldsymbol{\lambda}}^{N(k-1)+2}\left(\mathbf{z}_2, d\mathbf{z}_2'\right) \cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{N(k-1)+N}\left(\mathbf{z}_N, d\mathbf{z}_N'\right).$$

Now, the convergence in total variation $d_{\mathrm{TV}}(\cdot, \cdot)$ can be shown to decrease geometrically as

$$\begin{aligned}
&d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, \cdot\right), \Pi\right) \\
&\quad = \sup_{A} \left| \Pi(A) - P^{k}\left(\boldsymbol{\eta}, A\right) \right| && \textit{Definition of } d_{TV} \\
&\quad \leq \sup_{A} \Bigg| \int_{A} \pi\left(d\mathbf{z}'^{(1)}\right) \times \ldots \times \pi\left(d\mathbf{z}'^{(N)}\right) \\
&\qquad\qquad - K_{\boldsymbol{\lambda}}^{(k-1)N+1}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{(1)}\right) \times \ldots \times K_{\boldsymbol{\lambda}}^{kN}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{(N)}\right) \Bigg| \\
&\quad \leq \sup_{A} \sum_{n=1}^{N} \left| \int_{A} \pi\left(d\mathbf{z}^{(n)}\right) - K_{\boldsymbol{\lambda}}^{(k-1)N+n}\left(\mathbf{z}^{(n)}, d\mathbf{z}'^{(n)}\right) \right| && \textit{Lemma 1} \\
&\quad = \sum_{n=1}^{N} d_{\mathrm{TV}}\left(K_{\boldsymbol{\lambda}}^{(k-1)N+n}\left(\mathbf{z}^{(n)}, \cdot\right), \pi\right) && \textit{Definition of } d_{TV} \\
&\quad \leq \sum_{n=1}^{N} \rho^{(k-1)N+n} && \textit{Geometric ergodicity} \\
&\quad = \rho^{kN} \rho^{-N} \frac{\rho - \rho^{N+1}}{1-\rho} && \textit{Solved sum} \\
&\quad = \frac{\rho\left(1-\rho^{N}\right)}{\rho^{N}(1-\rho)} \left(\rho^{N}\right)^{k}.
\end{aligned}$$

Although the constant depends on ρ and N, the kernel P is geometrically ergodic and converges N times faster than the base kernel K.

Bound on the Gradient Variance To analyze the variance of the gradient, we require detailed information about the n-step marginal transition kernel, which is unavailable in general for most MCMC kernels. Fortunately, specifically for the IMH kernel, Smith & Tierney (1996) have shown that the n-step marginal IMH kernel is given as Equation (2). From this, we show that

$$\begin{aligned}
&\mathbb{E}\left[\left\| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \right\|_{*}^{2} \middle| \mathcal{F}_t \right] \\
&\quad = \mathbb{E}\left[\left\| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \right\|_{*}^{2} \middle| \mathbf{z}_{t-1}^{(N)}, \boldsymbol{\lambda}_{t-1} \right] \\
&\quad = \mathbb{E}\left[\left\| \frac{1}{N} \sum_{n=1}^{N} \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}^{(n)}\right) \right\|_{*}^{2} \middle| \mathbf{z}_{t-1}^{(N)}, \boldsymbol{\lambda}_{t-1} \right] \\
&\quad \leq \mathbb{E}\left[\frac{1}{N^2} \sum_{n=1}^{N} \left\| \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}^{(n)}\right) \right\|_{*}^{2} \middle| \mathbf{z}_{t-1}^{(N)}, \boldsymbol{\lambda}_{t-1} \right] && \textit{Triangle inequality} \\
&\quad = \frac{1}{N^2} \sum_{n=1}^{N} \mathbb{E}_{\mathbf{z}^{(n)} \sim K^{n}(\mathbf{z}_{t-1}, \cdot)} \left[\left\| \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}^{(n)}\right) \right\|_{*}^{2} \middle| \mathbf{z}_{t-1}^{(N)}, \boldsymbol{\lambda}_{t-1} \right] && \textit{Linearity of expectation} \\
&\quad \leq \frac{1}{N^2} \sum_{n=1}^{N} n \left(1 - \frac{1}{w^*}\right)^{n-1} \mathbb{E}_{\mathbf{z}^{(n)} \sim q_{\mathrm{def.}}(\cdot; \boldsymbol{\lambda})} \left[\left\| \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}^{(n)}\right) \right\|_{*}^{2} \right] \\
&\qquad + \left(1 - \frac{1}{w^*}\right)^{n} \left\| \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}_{t-1}^{(N)}\right) \right\|_{*}^{2} && \textit{Lemma 4}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{N^2}\sum_{n=1}^{N} n\left(1-\frac{1}{w^*}\right)^{n-1} L^2 + \left(1-\frac{1}{w^*}\right)^{n} L^2 && \textit{Assumption 3}\\
&= \frac{L^2}{N^2}\sum_{n=1}^{N} n\left(1-\frac{1}{w^*}\right)^{n-1} + \left(1-\frac{1}{w^*}\right)^{n} && \textit{Moved constant forward}\\
&= \frac{L^2}{N^2}\left[(w^*)^2 + w^* - \left(1-\frac{1}{w^*}\right)^{N}\left((w^*)^2 + w^* + N\,w^*\right)\right] && \textit{Solved sum}\\
&= \frac{L^2}{N^2}\left[\frac{1}{2}N^2 + \frac{3}{2}N + \mathcal{O}\left(1/w^*\right)\right] && \textit{Laurent series expansion at } w^* \to \infty\\
&= L^2\left[\frac{1}{2} + \frac{3}{2}\frac{1}{N} + \mathcal{O}\left(1/w^*\right)\right].
\end{aligned}
$$

The Laurent approximation becomes exact as $w^* \to \infty$, which is useful as a consequence of Proposition 2. □

Theorem 3. pMCSA, our proposed scheme, is obtained by setting

$$
P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, d\boldsymbol{\eta}'\right) = K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{\,(1)}\right) K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(2)}, d\mathbf{z}'^{\,(2)}\right)\cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{\,(N)}\right)
$$

with $\boldsymbol{\eta} = \left(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N)}\right)$. Then, given Assumption 2 and 3, the mixing rate and the gradient variance bounds are

$$
d_{\mathrm{TV}}\left(P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, \cdot\right), \Pi\right) \leq C\left(N\right)\rho^{k} \quad \text{and} \quad \mathbb{E}\left[\,\|\mathbf{g}\left(\boldsymbol{\lambda}, \boldsymbol{\eta}\right)\|_{*}^{2}\,\middle|\,\mathcal{F}_t\,\right] \leq L^2\left[\,\frac{1}{N} + \frac{1}{N}\left(1-\frac{1}{w^*}\right)\right],
$$

where $w^* = \sup_{\mathbf{z}} \pi\left(\mathbf{z}\right)/q_{\text{def.}}\left(\mathbf{z};\boldsymbol{\lambda}\right)$ and C is some positive constant depending on N.

Proof of Theorem 3. Our proposed scheme, pMCSA, is described in Algorithm 4. At each iteration, our scheme performs a single MCMC transition for each of the N samples, or chains, to estimate the gradient. Similarly to JSA, we use the IMH kernel $K_{\boldsymbol{\lambda}}$.

Ergodicity of the Markov Chain Since our kernel operates the same MCMC kernel $K_{\boldsymbol{\lambda}}$ for each of the N parallel Markov chains, the n-step marginal kernel $P_{\boldsymbol{\lambda}}$ can be represented as

$$
P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, d\boldsymbol{\eta}'\right) = K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(1)}, d\mathbf{z}'^{\,(1)}\right) K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(2)}, d\mathbf{z}'^{\,(2)}\right)\cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}^{(N)}, d\mathbf{z}'^{\,(N)}\right).
$$

Then, the convergence in total variation $d_{\mathrm{TV}}\left(\cdot,\cdot\right)$ can be shown to decrease geometrically as

$$
\begin{aligned}
&d_{\mathrm{TV}}\left(K_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, \cdot\right), \Pi\right)\\
&\quad= \sup_{A}\left|\,\Pi\left(A\right) - P_{\boldsymbol{\lambda}}^{k}\left(\boldsymbol{\eta}, A\right)\right| && \textit{Definition of } d_{TV}\\
&\quad\leq \sup_{A}\Big|\int_{A} \pi\left(d\mathbf{z}_1'\right)\cdot \ldots \cdot \pi\left(d\mathbf{z}_N'\right)\\
&\qquad\qquad - K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}_1, d\mathbf{z}_1'\right)\cdot \ldots \cdot K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}_N, d\mathbf{z}_N'\right)\,\Big|\\
&\quad\leq \sup_{A}\sum_{n=1}^{N}\left|\int_{A}\pi\left(d\mathbf{z}_k'\right) - K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}_n, d\mathbf{z}_n'\right)\right| && \textit{Lemma 1}\\
&\quad= \sum_{n=1}^{N} d_{\mathrm{TV}}\left(K_{\boldsymbol{\lambda}}^{k}\left(\mathbf{z}_n, \cdot\right), \pi\right) && \textit{Definition of TV distance}\\
&\quad\leq \sum_{n=1}^{N}\rho^{k} && \textit{Geometric ergodicity}\\
&\quad= N\,\rho^{k} && \textit{Solved sum.}
\end{aligned}
$$

Bound on the Gradient Variance The bound on the gradient variance can be derived in similar manner to JSA as

$$
\begin{aligned}
&\mathbb{E}\left[\| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \,\middle|\, \mathcal{F}_t \right] \\
&\quad = \mathbb{E}\left[\| \mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\eta}) \|_*^2 \,\middle|\, \mathbf{z}_{t-1}^{(1:N)}, \boldsymbol{\lambda}_{t-1} \right] \\
&\quad = \mathbb{E}\left[\left\| \frac{1}{N} \sum_{n=1}^{N} \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_n) \right\|_*^2 \,\middle|\, \mathbf{z}_{t-1}^{(1:N)}, \boldsymbol{\lambda}_{t-1} \right] \\
&\quad \leq \mathbb{E}\left[\frac{1}{N^2} \sum_{n=1}^{N} \| \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_n) \|_*^2 \,\middle|\, \mathbf{z}_{t-1}^{(1:N)}, \boldsymbol{\lambda}_{t-1} \right] && \textit{Triangle inequality} \\
&\quad = \frac{1}{N^2} \sum_{n=1}^{N} \mathbb{E}_{\mathbf{z}_n \sim K_{\boldsymbol{\lambda}_{t-1}}\left(\mathbf{z}_{t-1}^{(n)}, \cdot\right)} \left[\| \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_n) \|_*^2 \,\middle|\, \mathbf{z}_{t-1}^{(1:N)}, \boldsymbol{\lambda}_{t-1} \right] && \textit{Linearity of expectation} \\
&\quad \leq \frac{1}{N^2} \sum_{n=1}^{N} \mathbb{E}_{\mathbf{z}_n \sim q_{\text{def.}}(\cdot; \boldsymbol{\lambda}_{t-1})} \left[\| \mathbf{s}(\boldsymbol{\lambda}; \mathbf{z}_n) \|_*^2 \right] + \left(1 - \frac{1}{w^*}\right) \left\| \mathbf{s}\left(\boldsymbol{\lambda}; \mathbf{z}_{t-1}^{(n)}\right) \right\|_*^2 && \textit{Lemma 4} \\
&\quad \leq \frac{1}{N^2} \sum_{n=1}^{N} L^2 + \left(1 - \frac{1}{w^*}\right) L^2 && \textit{Assumption 3} \\
&\quad = \frac{L^2}{N^2} \sum_{n=1}^{N} 1 + \left(1 - \frac{1}{w^*}\right) && \textit{Moved constant forward} \\
&\quad = L^2 \left[\frac{1}{N} + \frac{1}{N} \left(1 - \frac{1}{w^*}\right) \right]. && \textit{Solved sum}
\end{aligned}
$$

□

E Additional Experimental Results

E.1 Bayesian Neural Network Regression

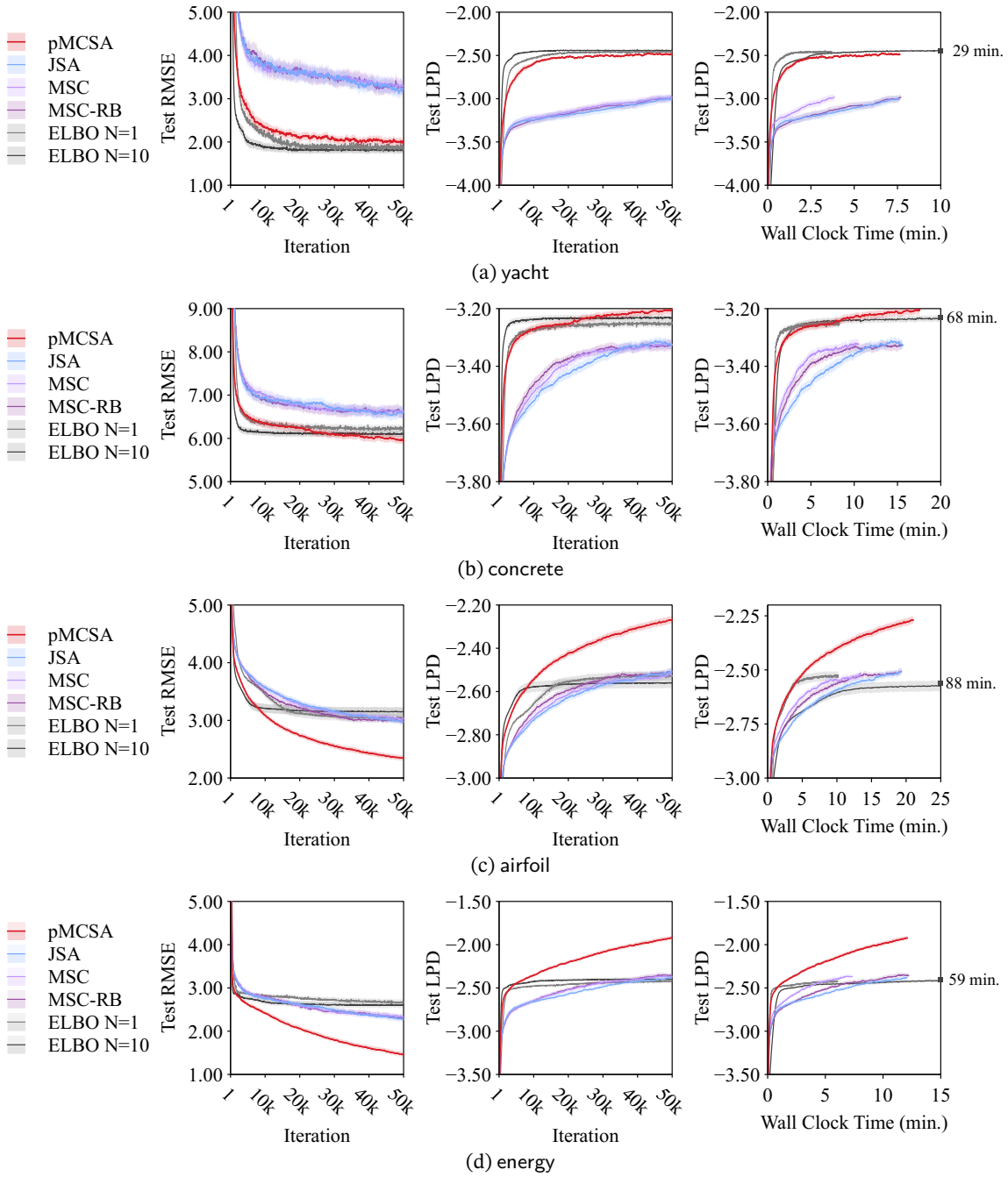

Figure 3: **Test root-mean-square error (RMSE) and test log predictive density (LPD) on Bayesian neural network regression.** The grey squares (■) mark the performance of ELBO $N = 10$ at the wall clock time shown next to it. The error bands show the 95% bootstrap confidence intervals obtained from 20 independent 90% train-test splits.

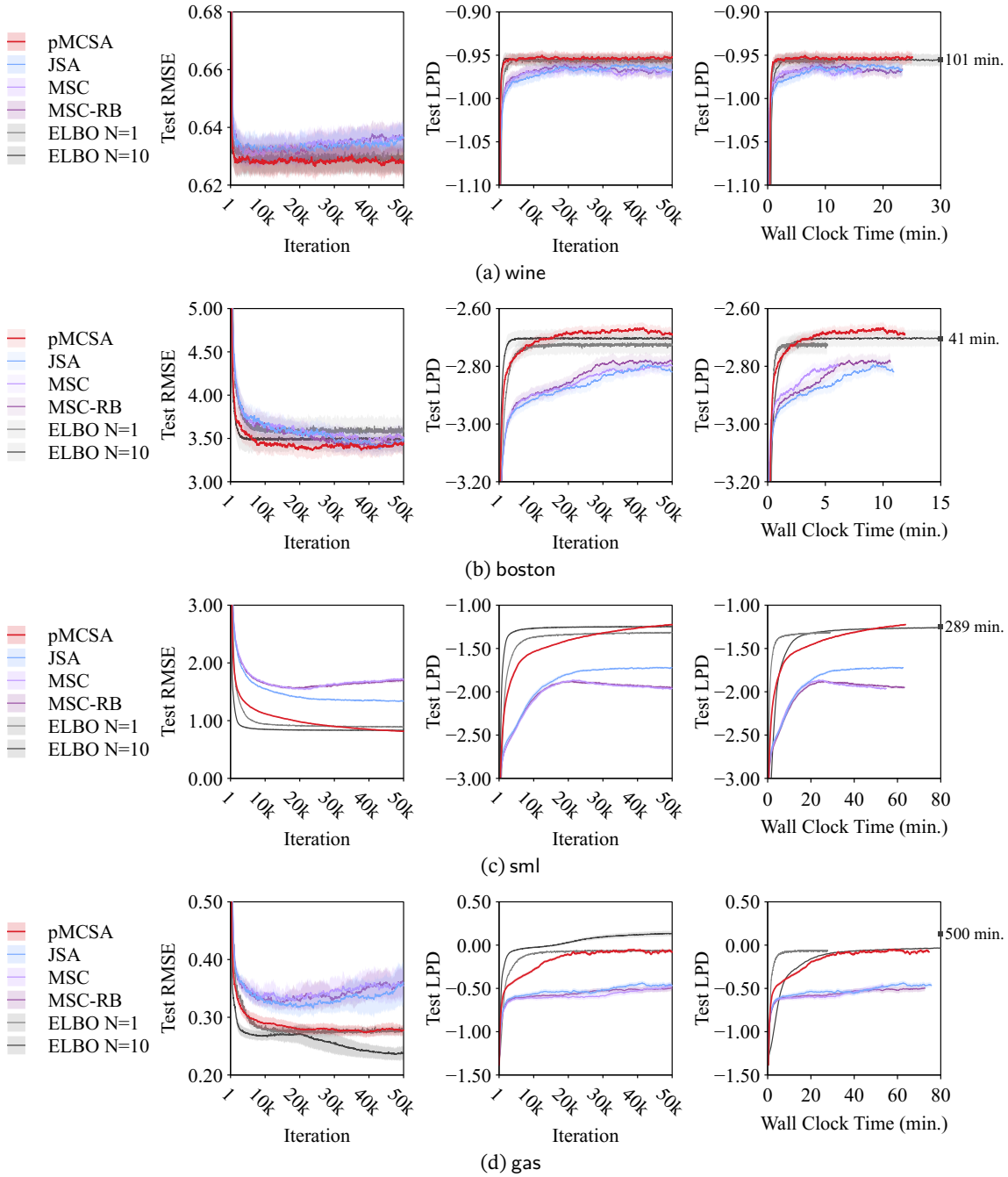

Figure 4: **(continued) Test root-mean-square error (RMSE) and test log predictive density (LPD) on Bayesian neural network regression.** The grey squares (■) mark the performance of ELBO $N = 10$ at the wall clock time shown next to it. The error bands show the 95% bootstrap confidence intervals obtained from 20 independent 90% train-test splits.

E.2 Robust Gaussian Process Regression

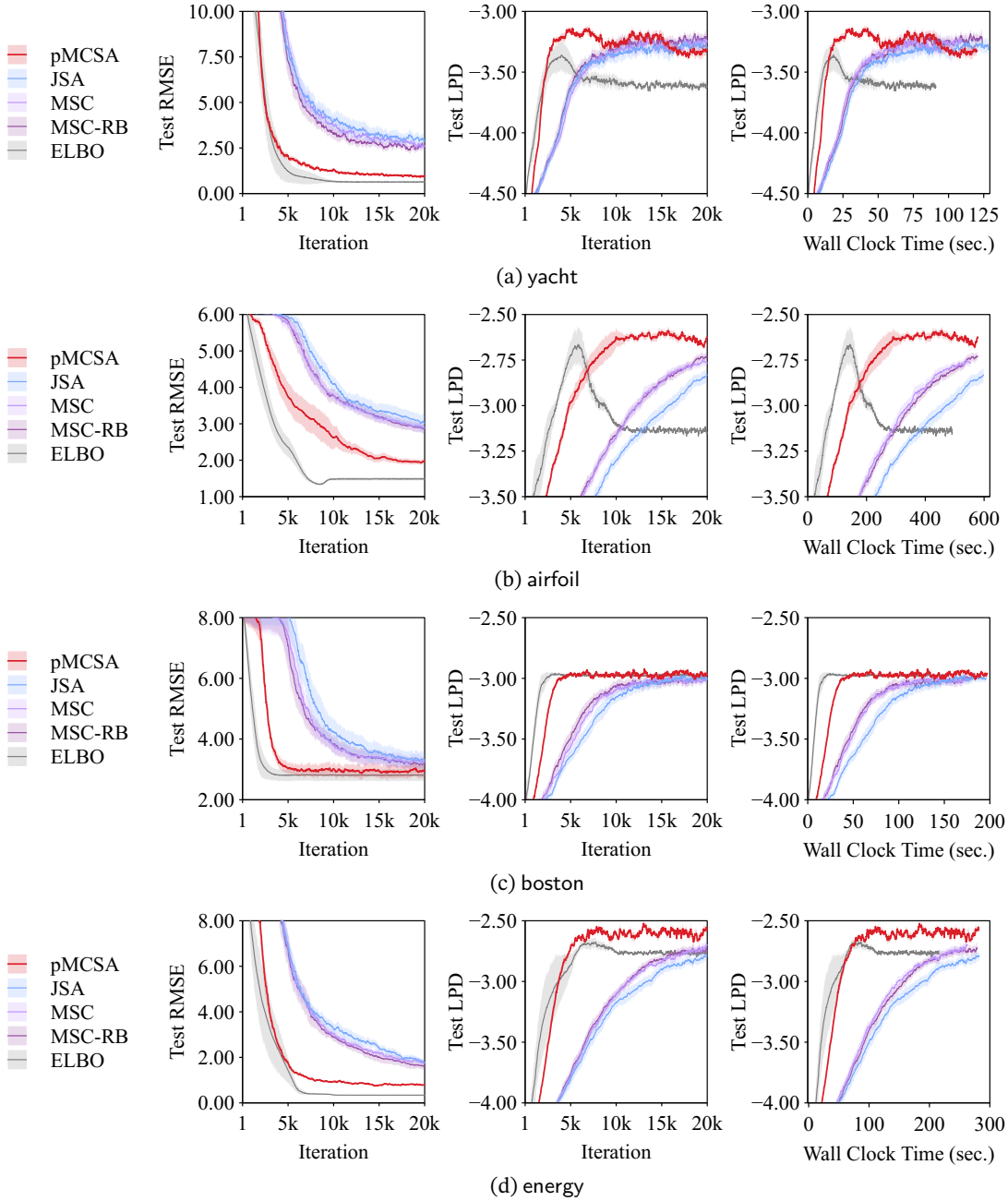

Figure 5: **Test root-mean-square error (RMSE) and test log predictive density (LPD) on robust Gaussian process regression.** The error bands shows the 95% bootstrap confidence interval obtained from 20 repetitions.

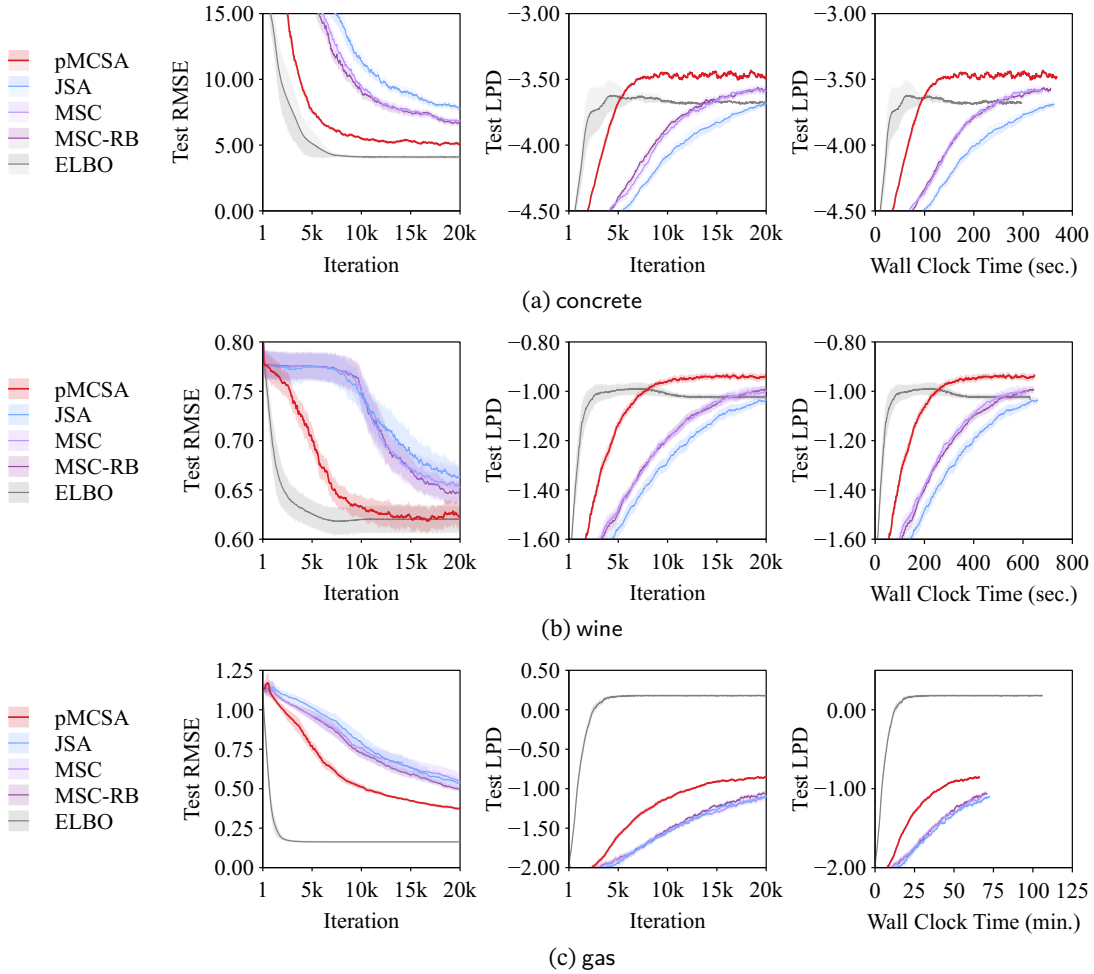

Figure 6: **(continued) Test root-mean-square error (RMSE) and test log predictive density (LPD) on robust Gaussian process regression.** The error bands show the 95% bootstrap confidence intervals obtained from 20 independent 90% train-test splits.