Journal of Machine Learning Research 25 (2024) 1-71 Submitted 3/22; Revised 4/24; Published 7/24

A Framework for Improving the Reliability of Black-box Variational Inference

Manushi Welandawe MANUSHIW@BU.EDU
Department of Mathematics & Statistics
Boston University, USA

Michael Riis Andersen MIRI@DTU.DK
DTU Compute
Technical University of Denmark, Denmark

Aki Vehtari AKI.VEHTARI@AALTO.FI
Department of Computer Science
Aalto University, Finland

Jonathan H. Huggins HUGGINS@BU.EDU
Department of Mathematics & Statistics
Faculty of Computing & Data Sciences
Boston University, USA

Editor: Justin Domke

Abstract

Black-box variational inference (BBVI) now sees widespread use in machine learning and statistics as a fast yet flexible alternative to Markov chain Monte Carlo methods for approximate Bayesian inference. However, stochastic optimization methods for BBVI remain unreliable and require substantial expertise and hand-tuning to apply effectively. In this paper, we propose *robust and automated black-box VI* (RABVI), a framework for improving the reliability of BBVI optimization. RABVI is based on rigorously justified automation techniques, includes just a small number of intuitive tuning parameters, and detects inaccurate estimates of the optimal variational approximation. RABVI adaptively decreases the learning rate by detecting convergence of the fixed–learning-rate iterates, then estimates the symmetrized Kullback–Leibler (KL) divergence between the current variational approximation and the optimal one. It also employs a novel optimization termination criterion that enables the user to balance desired accuracy against computational cost by comparing (i) the predicted relative decrease in the symmetrized KL divergence if a smaller learning were used and (ii) the predicted computation required to converge with the smaller learning rate. We validate the robustness and accuracy of RABVI through carefully designed simulation studies and on a diverse set of real-world model and data examples.

Keywords: black-box variational inference, symmetrized KL divergence, stochastic optimization, fixed-learning rate

1. Introduction

A core strength of the Bayesian approach is that it is conceptually straightforward to carry out inference in *arbitrary* models, which enables the user to employ whatever model is most appropriate for the problem at hand. The flexibility and uncertainty quantification provided

by Bayesian inference have led to its widespread use in statistics (Robert, 2007; Gelman et al., 2013) and machine learning (Bishop, 2006; Murphy, 2012), including in deep learning (for example, Kingma and Welling, 2014; Rezende et al., 2014; Gal and Ghahramani, 2016; Maddox et al., 2019; Saatchi and Wilson, 2017; Johnson et al., 2016; Burda et al., 2016). Using Bayesian methods in practice, however, typically requires using approximate inference algorithms to estimate quantities of interest such as posterior functionals (for example, means, covariances, predictive distributions, and tail probabilities) and measures of model fit (for example, marginal likelihoods and cross-validated predictive accuracy). Therefore, a core challenge in modern Bayesian statistics is the development of *general-purpose* (or *black-box*) algorithms that can accurately approximate these quantities for whatever model the user chooses. In machine learning, rather than using Markov chain Monte Carlo (MCMC), *black-box variational inference* (BBVI) has become the method of choice because of its scalability and wide-applicability (Wainwright and Jordan, 2008; Blei et al., 2017; Kingma and Welling, 2014; Rezende et al., 2014; Burda et al., 2016). BBVI is implemented in many software packages for general-purpose inference such as Stan, Pyro, PyMC3, and TensorFlow Probability, which have seen widespread adoption by applied data analysts, statisticians, and data scientists.

Variational inference methods aim to minimize a measure of discrepancy between a parameterized family of distributions and the posterior distribution, with the Kullback–Leibler divergence being the canonical choice of discrepancy. Conventional approaches to variational inference leverage conditional conjugacy and other model-specific structure to derive optimization algorithms. BBVI, on the other hand, uses stochastic optimization to avoid the need for model-specific derivations, thereby significantly broadening the applicability of variational methods. Ensuring the efficiency and reliability of the BBVI optimization requires careful selection of optimization method and the stochastic estimator of the discrepancy gradient. For example, using adaptive optimization procedures like Adam, RMSProp, and Adagrad can ensure stability (Hinton and Tieleman, 2012; Kingma and Ba, 2015; Duchi et al., 2011). Due to its relatively small variance, the most common gradient estimator is the reparameterization gradient (Salimans and Knowles, 2013; Kingma and Welling, 2014; Rezende et al., 2014; Ruiz et al., 2016; Bamler et al., 2017; Domke, 2019). However, sometimes alternatives such as the score function gradient are employed (Paisley et al., 2012; Ranganath et al., 2014). No matter the choice of gradient estimator, various variance reduction strategies like control variates have been introduced to stabilize and speed up optimization (Miller et al., 2017; Roeder et al., 2017; Geffner and Domke, 2018, 2020; Boustati et al., 2020; Wang et al., 2023a,b). Recent research has furthered our understanding of convergence behavior, tackling theoretical challenges in stochastic optimization and providing new convergence guarantees (Domke et al., 2023; Kim et al., 2023).

While some recent progress has been made in developing tools for assessing the accuracy of variational approximations (Yao et al., 2018; Huggins et al., 2020; Wang et al., 2023c), stochastic optimization methods for BBVI remain unreliable and require substantial hand-tuning of the number of iterations and optimizer tuning parameters. Moreover, there are few tools available for determining whether the variational parameters estimated by these frameworks are close to optimal in any meaningful sense and, if not, how to address the problem; More iterations? A different learning rate schedule? A smaller initial or final learning rate? Agrawal et al. (2020) demonstrate the absence of a reliable and coherent

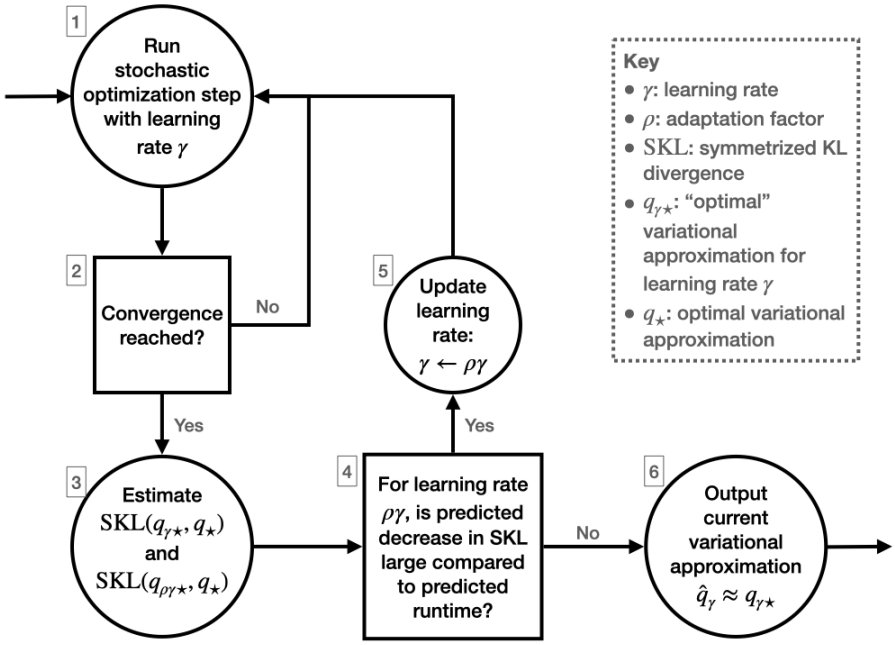
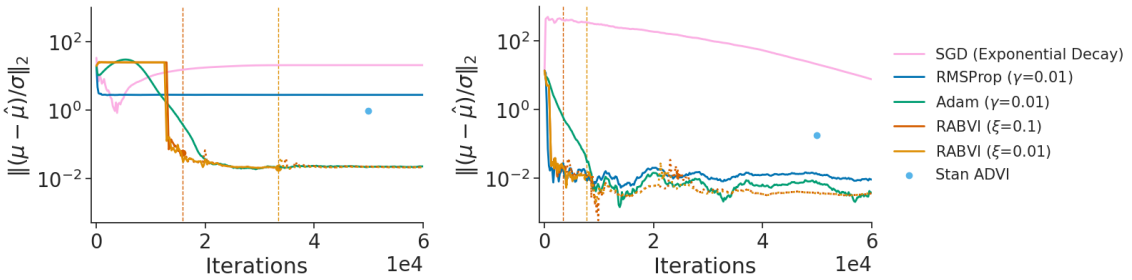


Figure 1: Schematic of the high-level logic of our proposed *robust and automated BBVI* (RABVI) framework.

optimization methodology for BBVI. The authors synthesize and compare recent advances such as normalizing flows and gradient estimators using 30 benchmarked models. Despite the fact that these models vary greatly in the complexity and dimensionality of the posteriors, Agrawal et al. (2020) run each optimization for a fixed number of iterations (30,000) and for 5 different step sizes because the existing literature does not provide any compelling guidance for how to automate the choice of step size and reliably determine when the optimization has converged.

1.1 Contributions

In view of the significant limitations of existing BBVI optimization methodology, in this paper we aim to provide a practical, cohesive, and theoretically well-grounded optimization framework for BBVI. To ensure reliability and wide applicability, we develop a framework that is (1) automated, (2) intuitively adjustable by the user, and (3) robust to failure and tuning parameter selection. Our approach builds on a recent line of work inspired by Pflug (1990), which uses a fixed learning rate γ that is adaptively decreased by a multiplicative factor ρ once the optimization iterates, which form a homogenous Markov chain, have converged (Chee and Toulis, 2018; Yaida, 2019; Pesme et al., 2020; Chee and Li, 2020; Zhang et al., 2020; Dhaka et al., 2020). A benefit of this approach is that, for a given learning rate, a dramatically more accurate estimate of the optimal variational parameter



(a) *nes2000* (b) *dogs*

Figure 2: Accuracy comparison of variational inference algorithms using two data sets/models from the `posteriordb` package (see Section 7.2 for details). Accuracy is measured in terms of relative mean error $\|(\mu - \hat{\mu})/\sigma\|_2$, where μ and σ are, respectively, the posterior mean and standard deviation vectors and $\hat{\mu}$ is the variational approximation to μ. The vertical lines indicate the termination points for RABVI, which uses averaged Adam (see Section 4.3). The fixed-learning rate algorithms have a learning rate γ and RABVI has a user-specified accuracy threshold ξ. For SGD exponential decay, we use an initial learning rate of 0.01 for *dogs* but a smaller initial learning rate of 0.001 for *nes2000* due to optimization instability.

can be obtained by using *iterate averaging* (Ruppert, 1988; Polyak and Juditsky, 1992; Dieuleveut et al., 2020). However, as we have shown in previous work, existing convergence checks can be unreliable and stop too early (Dhaka et al., 2020). Since the learning rate is decreased by a constant multiplicative factor, decreasing it too early can slow down the optimization by an order of magnitude or more. Hence, it is crucial to develop methods that do not prematurely declare convergence. On the other hand, an optimization framework must also provide a termination criterion that triggers when it is no longer worthwhile to decrease the learning rate further, either because the current variational approximation is sufficiently accurate or because further optimization would be too time-consuming.

The key idea that informs our solutions to these challenges is that we want $q_{\gamma *}$, the target variational approximation for learning rate γ, to be close to the optimal variational approximation q_*. We measure closeness in terms of *symmetrized Kullback–Leibler (KL) divergence* $\mathrm{SKL}(q_*, q_{\gamma *})$ and show that closeness in symmetrized KL divergence can be translated into bounds on other widely used accuracy metrics like Wasserstein distance (Huggins et al., 2020; Bolley and Villani, 2005). Figure 1 summarizes our proposed framework, which we call *robust and automated black-box VI (RABVI)*. The primary contributions of this paper are in steps 2, 3, and 4. In step 2, to determine convergence at a fixed step size, we build upon our approach in Dhaka et al. (2020), where we establish that the scale-reduction factor $\widehat{R}$ (Gelman and Rubin, 1992; Gelman et al., 2013; Vehtari et al., 2021), which is widely used to determine convergence of Markov chain Monte Carlo algorithms, can be combined with a Monte Carlo standard error (MCSE) (Geyer, 1992; Vehtari et al., 2021) cutoff to construct a convergence criteria. We improve upon our previous proposal by:

(a) adaptively finding the size of the convergence window, which may need to be large for challenging or high-dimensional distributions over the model parameters, and

(b) developing a new rigorous MCSE cutoff condition that guarantees the symmetrized KL divergence between $q_{\rho\gamma*}$ and the estimate of $q_{\rho\gamma*}$ obtained via iterate averaging will be small.

In step 3, we leverage recent results that characterize the bias of the stationary distribution of stochastic gradient descent (SGD) with a fixed learning rate (Dieuleveut et al., 2020) to estimate $\mathrm{SKL}(q_*, q_{\gamma*})$ and $\mathrm{SKL}(q_*, q_{\rho\gamma*})$ without access to q_*. In step 4, these estimates enable the use of a termination criterion that compares (i) the predicted relative decrease in the KL divergence if the smaller learning $\rho\gamma$ were used and (ii) the predicted computation required to converge with the learning rate $\rho\gamma$. By trading off between (i) and (ii), the criterion enables the user to balance desired accuracy against computational cost. Figure 2 provides an example of the faster convergence, higher accuracy, and greater reliability achievable using RABVI compared to alternative optimization algorithms and demonstrates how the user can trade off accuracy and computation by adjusting the accuracy threshold ξ.

In summary, by drawing on recent developments in theory and methods for fixed–learning-rate stochastic optimization, tools from MCMC methodology and results from functional analysis, RABVI delivers a number of benefits:

- it relies on rigorously justified automation techniques, including automatic learning rate adaptation;
- it has an interpretable, user-adjustable accuracy parameter along with a small number of additional intuitive tuning parameters;
- it detects inaccurate estimates of the optimal variational approximation; and
- it can flexibly incorporate additional or future methodological improvements related to variational inference and stochastic optimization.

We demonstrate through synthetic comparisons and real-world model and data examples that RABVI provides high-quality black-box approximate inferences. We make RABVI available as part of the open source Python package VIABEL.[1]

2. Preliminaries and Background

In this section, we briefly review relevant background about Bayesian and variational inference.

2.1 Bayesian Inference

Let $\theta \in \mathbb{R}^d$ denote a parameter vector of interest, and let x denote observed data. A Bayesian model consists of a prior density $\pi_0(\mathrm{d}\theta)$ and a likelihood $\ell(x;\theta)$. Together, the prior and likelihood define a joint distribution over the data and parameters. The Bayesian posterior

1. `https://github.com/jhuggins/viabel`

distribution π is the conditional distribution of θ given fixed data x, with x suppressed in the notation since it is always fixed throughout this work. To write this conditional, we define the unnormalized posterior density $\pi^u(\theta) := \ell(x;\theta)\pi_0(\mathrm{d}\theta)$ and the marginal likelihood, or evidence, $Z := \int \pi^u(\mathrm{d}\theta)$. Then the posterior is $\pi := \pi^u/Z$. Typically, practitioners report *posterior summaries*, such as point estimates and uncertainties, rather than the full posterior. For $\vartheta \sim \pi$, typical summaries include the mean $m_\pi := \mathbb{E}(\vartheta)$, the covariance $\Sigma_\pi := \mathbb{E}\{(\vartheta - m_\pi)(\vartheta - m_\pi)^\top\}$, and $[a,b]$ interval probability $I_{\pi,i,a,b} := \mathbb{P}(\vartheta_i \in [a,b]) = \mathbb{E}\{\mathbb{1}(\vartheta_i \in [a,b])\}$, where $\mathbb{1}(C)$ is equal to one when C is true and zero otherwise.

2.2 Variational Inference

In most Bayesian models, it is infeasible to efficiently compute quantities of interest such as posterior means, variances, and quantiles. Therefore, one must use an approximate inference method that produces an approximation q to the posterior π. The summaries of q may in turn be used as approximations to the summaries of π. One approach, *variational inference*, is widely used in machine learning. Variational inference aims to minimize some measure of discrepancy $\mathcal{D}_\pi(\cdot)$ over a tractable family $\mathcal{Q} = \{q_\lambda : \lambda \in \mathbb{R}^m\}$ of approximating distributions (Wainwright and Jordan, 2008; Blei et al., 2017):

$$q_{\lambda_*} = \operatorname*{arg\,min}_{q_\lambda \in \mathcal{Q}} \mathcal{D}_\pi(q_\lambda).$$

The variational family $\mathcal{Q}$ is chosen to be tractable in the sense that, for any $q \in \mathcal{Q}$, we are able to efficiently calculate relevant summaries either analytically or using independent and identically distributed samples from q.

In variational inference, the standard choice for the discrepancy $\mathcal{D}_\pi(\cdot)$ is the *Kullback–Leibler (KL) divergence* $\mathrm{KL}(q \mid \pi) := \int \log(\mathrm{d}q/\mathrm{d}\pi)\,\mathrm{d}q$ (Bishop, 2006). Note that the KL divergence is asymmetric in its arguments. The direction $\mathcal{D}_\pi(q) = \mathrm{KL}(q \mid \pi)$ is most typical in variational inference, largely out of convenience; the unknown marginal likelihood Z appears in an additive constant that does not influence the optimization and computing gradients require estimating expectations only with respect to $q \in \mathcal{Q}$, which is chosen to be tractable. Minimizing $\mathrm{KL}(q \mid \pi)$ is equivalent to maximizing the *evidence lower bound* (ELBO; Bishop, 2006):

$$\mathsf{ELBO}(q) := \int \log\left(\frac{\mathrm{d}\pi^u}{\mathrm{d}q}\right)\mathrm{d}q.$$

While numerous other divergences have been used in the literature (for example, Hernández-Lobato et al., 2016; Li and Turner, 2016; Bui et al., 2017; Dieng et al., 2017; Wang et al., 2018; Wan et al., 2020), we focus on $\mathrm{KL}(q \mid \pi)$ since it is the most common choice; the default or only choice in widely used frameworks such as Stan, Pyro, and PyMC3; and easiest to estimate when using simple Monte Carlo sampling to approximate the gradient (Dhaka et al., 2021).

2.3 Black-box Variational Inference

Black-box variational inference (BBVI) methods have greatly extended the applicability of variational inference by removing the need for model-specific derivations (Cornebise et al.,

2008; Ranganath et al., 2014; Kucukelbir et al., 2015; Titsias and Lázaro-Gredilla, 2014; Mohamed et al., 2020) and enabling the use of more flexible approximation families (Kingma and Welling, 2014; Salimans et al., 2015; Papamakarios et al., 2021). This flexibility is a result of using simple Monte Carlo (and automatic differentiation) to approximate the (gradients of the) expectations that define common choices of the discrepancy objective (Papamakarios et al., 2021; Mohamed et al., 2020). To estimate the optimal variational parameter λ_*, BBVI methods commonly use stochastic optimization schemes which at iteration k are of the form

$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \gamma^{(k)} d^{(k)}, \tag{1}$$

where $d^{(k)} \in \mathbb{R}^m$ is the *descent direction* and $\gamma^{(k)} > 0$ is the *learning rate* (also called the *step size*). Standard stochastic gradient descent corresponds to taking $d^{(k)} = \hat{g}^{(k)}$, a (usually unbiased) stochastic estimate of the gradient $g(\lambda^{(k)}) := \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}})$. We are particularly interested in adaptive stochastic optimization methods (for example, Duchi et al., 2011; Hinton and Tieleman, 2012; Kingma and Ba, 2015) that use a smoothed and/or rescaled version of $\hat{g}^{(k)}$ as the descent direction. For example, RMSProp (Hinton and Tieleman, 2012) tracks an exponential moving average of the squared gradient, $\nu^{(k+1)} = \beta\nu^{(k)} + (1-\beta)\hat{g}^{(k)} \odot \hat{g}^{(k)}$, which is used to rescale the current stochastic gradient: $d^{(k)} = \hat{g}^{(k+1)}/\sqrt{\nu^{(k)}}$. Or, Adam (Kingma and Ba, 2015) tracks an exponential moving average of the gradient $m^{(k+1)} = \alpha m^{(k)} + (1-\alpha)\hat{g}^{(k)}$ as well as the squared gradient $\nu^{(k+1)}$ and uses both to rescale the current stochastic gradient: $d^{(k)} = m^{(k)}/\sqrt{\nu^{(k)}}$. The benefits of adaptive algorithms include that they tend to be more stable and are scale invariant, so the learning rate can be set in a problem-independent manner.

2.4 Fixed–Learning-Rate Stochastic Optimization

If the learning rate is fixed so that $\gamma^{(k)} = \gamma$, then we can view the iterates $\lambda^{(1)}, \lambda^{(2)}, \ldots$ produced by Eq. (1) as a homogenous Markov chain, which under certain conditions will have a stationary distribution μ_γ (Dieuleveut et al., 2020; Gitman et al., 2019; Pflug, 1990; Chee and Toulis, 2018; Yaida, 2019). Stochastic optimization with a fixed learning rate exhibits two distinct phases: a transient (a.k.a. warm-up) phase during which iterates make rapid progress toward the optimum, followed by a stationary (a.k.a. mixing) phase during which iterates oscillate around the mean of the stationary distribution, $\bar{\lambda}_\gamma := \int \lambda \, \mu_\gamma(\mathrm{d}\lambda)$ (Gelman and Rubin, 1992).

The mean $\bar{\lambda}_\gamma$ is a natural target because even if the variance of the individual iterates $\lambda^{(k)}$ means they are far from λ_*, $\bar{\lambda}_\gamma$ can be a much more accurate approximation to λ_*. For example, the following result quantifies the bias of standard fixed–learning-rate SGD (Gitman et al. (2019) provide similar results for momentum-based SGD algorithms):

Theorem 1 (Dieuleveut et al. (2020, Theorem 4)) *Under regularity conditions on the objective function and the unbiased stochastic gradients, there exist constant vectors $A, B \in \mathbb{R}^m$ such that*[2]

$$\bar{\lambda}_\gamma - \lambda_* = A\gamma + B\gamma^2 + o(\gamma^2) \tag{2}$$

2. As stated in Dieuleveut et al. (2020), the $B\gamma^2$ term is written as $O(\gamma^2)$. However, the fact that this term is of the form $B\gamma^2 + o(\gamma^2)$ can be extracted from the proof.

and a matrix $A' \in \mathbb{R}^{m \times m}$ *such that*

$$\int (\lambda - \lambda_*)(\lambda - \lambda_*)^\top \mu_\gamma(\mathrm{d}\lambda) = A'\gamma + O(\gamma^2).$$

Remark 2 *The regularity conditions required by Theorem 1 are mostly mild. For example, the stochastic gradients must be unbiased and have finite variance that does not grow too quickly away from the optimum. However, it does require the stronger assumptions that the objective function is smooth and strongly convex. While these conditions do not hold globally, we do not view it be a significant problem in practice because near the optimum we expect the objective function to be locally smooth and strongly convex.*

Theorem 1 shows that, at stationarity, a single iterate will satisfy $\lambda^{(k)} - \lambda_* = O(\gamma^{1/2})$ (with high probability) while its expectation will satisfy $\bar{\lambda}_\gamma - \lambda_* = O(\gamma)$. Therefore, when the learning rate is small, $\bar{\lambda}_\gamma$ is a substantially better estimator for λ_* than $\lambda^{(k)}$. In practice the *iterate average* (that is, the sample mean)

$$\hat{\lambda}_\gamma := \frac{1}{k^{\mathrm{avg}}} \sum_{k=0}^{k^{\mathrm{avg}}-1} \lambda^{(k^{\mathrm{conv}}+k)} \tag{3}$$

provides an estimate of $\bar{\lambda}_\gamma$, where k^{conv} is the iteration at which the optimization has reached the stationary phase and k^{avg} is the number of iterations used to compute the average. Using $\hat{\lambda}_\gamma$ as an estimate of λ_* is known as Polyak–Ruppert averaging (Polyak and Juditsky, 1992; Ruppert, 1988; Bach and Moulines, 2013).

When using iterate averaging, it is crucial to ensure the iterate average accurately approximates $\bar{\lambda}_\gamma$. Considering a stationary Markov chain, we can compute the Monte Carlo estimate of the mean of the Markov Chain at stationarity. Then the notion of *effective sample size* (ESS) aids in quantifying the accuracy of this Monte Carlo estimate. Further, the effective sample size can also be used to define the *Monte Carlo standard error* (MCSE) when the Markov chain satisfies a central limit theorem. We can efficiently estimate the ESS and also approximate the MCSE (see Appendix B.1 for details). We denote the estimates of ESS and MCSE for the ith component using iterates $\lambda_i^{(k^{\mathrm{conv}}:k)}$ as $\widehat{\mathrm{ESS}}(\lambda_i^{(k^{\mathrm{conv}}:k)})$ and $\widehat{\mathrm{MCSE}}(\lambda_i^{(k^{\mathrm{conv}}:k)})$ respectively. Dhaka et al. (2020) use the conditions $m^{-1}\sum_{i=1}^m \widehat{\mathrm{MCSE}}(\lambda_i^{(k^{\mathrm{conv}}:k)}) < 0.02$ and $\widehat{\mathrm{ESS}}(\lambda_i^{(k^{\mathrm{conv}}:k)}) > 20$ to determine when to stop iterate averaging. However, no rigorous justification is given for the MCSE threshold.

2.5 Automatically Scheduling Learning Rate Decreases

A benefit of using a fixed learning rate is that it can be adaptively and automatically decreased once the iterates reach stationarity. For example, if the initial learning rate is γ_0, after reaching stationarity the learning rate can decrease to $\gamma_1 := \rho\gamma_0$, where $\rho \in (0, 1)$ is a user-specified adaptation factor. The process can be repeated: when stationarity is reached at learning rate γ_t, the learning rate can decrease to $\gamma_{t+1} := \rho\gamma_t$. In this way the learning rate is not decreased too early (when the iterates are still making fast progress toward the optimum) or too late (when the accuracy of the iterates is no longer improving). Compare this adaptive approach to the canonical one of setting a schedule such as $\gamma^{(k)} = \triangle/(\bigcirc + k)^{\square}$,

which requires the choice of three tuning parameters. These tuning parameters can have a dramatic effect on the speed of convergence, particularly when $\lambda^{(0)}$ is far from λ_*.

The question of how to determine when the stationary phase has been reached has a long history with recent renewed attention (Pesme et al., 2020; Zhang et al., 2020; Chee and Li, 2020; Lang et al., 2019; Yaida, 2019; Pflug, 1990; Chee and Toulis, 2018; Dhaka et al., 2020). One line of work (Yaida, 2019; Lang et al., 2019; Zhang et al., 2020) is based on finding an invariant function that has expectation zero under the stationary distribution of the iterates, then using a test for whether the empirical mean of the invariant function is sufficiently close to zero. An alternative approach developed in Dhaka et al. (2020) makes use of the potential scale reduction factor $\widehat{R}$, perhaps the most widely used MCMC diagnostic for detecting stationarity (Gelman and Rubin, 1992; Gelman et al., 2013; Vehtari et al., 2021). The standard approach to computing $\widehat{R}$ is to use multiple Markov chains. If we have $J \geq 2$ chains and $K \gg 1$ iterates in each chain, then $\widehat{R} := (\hat{\mathbb{V}}/\hat{\mathbb{W}})^{1/2}$, where $\hat{\mathbb{V}}$ and $\hat{\mathbb{W}}$ are estimates of, respectively, the between-chain and within-chain variances. In the split-$\widehat{R}$ version, each chain is split into two before carrying out the computation above, which helps with detecting non-stationarity (Gelman et al., 2013; Vehtari et al., 2021) and allows for use even when $J = 1$. Let $\widehat{R}_i(W)$ denote the split-$\widehat{R}$ value computed from $\lambda_i^{(k-W+1)}, \ldots, \lambda_i^{(k)}$, the ith dimension of the last W iterates. Dhaka et al. (2020) uses the stationarity condition $\max_i \widehat{R}_i(100) < 1.1$,

3. Methodological Criteria

We now summarize our criteria when designing a robust and automatic optimization framework for BBVI.

Robustness. A robust method should not be too sensitive to the choice of tuning parameters. It should also work well on a wide range of "typical" problems. To achieve this we design an adaptive methods for setting parameters (such as the window size for detecting convergence) that are problem-dependent.

Automation. An automatic method should require minimal input from the user. Any inputs that are required should be clearly necessary (for example, the model and the data) or be intuitive to an applied user who is not an expert in variational inference and optimization. Therefore, we require the parameters of any adaptation scheme to either be intuitive or not require adjustment by the user. Examples of intuitive parameters include the maximum number of iterations, maximum runtime, and, when defined appropriately, accuracy.

We ensure these criteria are satisfied when designing the two core components of a BBVI stochastic optimization framework with automated learning rate scheduling:

1. A **termination rule** for stopping the optimization once the final approximation is close to the optimal approximation (Section 4).

2. A **learning rate scheduler**, which must detect stationarity and determine how many iterates to average before decreasing the learning rate (Section 5).

4. Termination Rule

The development of our termination rule will proceed in three steps. First, we will select an appropriate discrepancy measure between distributions. Next, we will design an idealized termination rule based on this discrepancy measure. Finally, we will develop an implementable version of the idealized termination rule that satisfies the criteria from Section 3.

4.1 Choice of Accuracy Measure

To develop a termination rule, we must specify a measure of how close a variational approximation returned by the optimization algorithm, $\hat{q}_*$, is to the optimal variational approximation $q_* := q_{\lambda_*}$. But the answer to this question depends upon choosing an appropriate measure of the discrepancy between q_* and the posterior π. Based on the discussion in Section 2.1, the goal should be for quantities such as m_{q_*}, Σ_{q_*}, and $I_{q_*,i,a,b}$ to be close to, respectively, m_π, Σ_π, and $I_{\pi,i,a,b}$. The interval probabilities are already on an interpretable scale, so ensuring that $|I_{q,i,a,b} - I_{\pi,i,a,b}|$ is much less than 1 is an intuitive notion of accuracy. Since $\Sigma_\pi^{1/2}$ establishes the relevant scale of the problem for means and standard deviations, so it is appropriate to ensure that $\|\Sigma_\pi^{-1/2}(m_\pi - m_{q_*})\|_2$ and $\|\Sigma_\pi^{-1}(\Sigma_\pi - \Sigma_{q_*})\|_2 = \|I - \Sigma_\pi^{-1}\Sigma_{q_*}\|_2$ are much less than 1.

While we want to choose a discrepancy measure that guarantees the accuracy of mean, covariance, and interval probabilities, ideally it would also guarantee other plausible expectations of interest (for example, predictive densities) are accurately approximated. The *Wasserstein distance* provides one convenient metric for accomplishing this goal, and is widely used in the analysis of MCMC algorithms and in large-scale data asymptotics (for example, Joulin and Ollivier, 2010; Madras and Sezer, 2010; Rudolf and Schweizer, 2018; Durmus et al., 2019; Durmus and Moulines, 2019; Vollmer et al., 2016; Eberle and Majka, 2019). For $p \geq 1$ and a positive-definite matrix $\Sigma \in \mathbb{R}^{d\times d}$, we define the (p, Σ)-*Wasserstein distance* between distributions η and ζ as

$$\mathcal{W}_{p,\Sigma}(\eta, \zeta) := \inf_{\omega} \left\{ \int \|\Sigma^{-1/2}(\theta - \theta')\|_2^p \omega(\mathrm{d}\theta, \mathrm{d}\theta') \right\}^{1/p},$$

where the infimum is over the set of *couplings* between η and ζ; that is, Borel measures ω on $\mathbb{R}^d \times \mathbb{R}^d$ such that $\eta = \omega(\cdot, \mathbb{R}^d)$ and $\zeta = \omega(\mathbb{R}^d, \cdot)$ (Villani, 2009, Defs. 6.1 & 1.1). Small (p, Σ)-Wasserstein distance implies many functionals of the two distributions are close relative to the scale determined by $\Sigma^{1/2}$.

Specifically, we have the following result, which is an immediate corollary of Huggins et al. (2020, Theorem 3.4).

Proposition 3 *If $\mathcal{W}_{p,\Sigma}(\eta, \zeta) \leq \varepsilon$ for any $p \geq 1$, then*

$$\|\Sigma^{-1/2}(m_\eta - m_\zeta)\|_2 \leq \varepsilon$$

If $\mathcal{W}_{2,\Sigma}(\eta, \zeta) \leq \varepsilon$, then, for $\varrho := \min\{\|\Sigma^{-1}\Sigma_\eta\|_2^{1/2}, \|\Sigma^{-1}\Sigma_\zeta\|_2^{1/2}\}$,

$$\|\Sigma^{-1}(\Sigma_\eta - \Sigma_\zeta)\|_2 < 2\varepsilon(\varrho + \varepsilon).$$

More generally, small (p, Σ)-Wasserstein distance for any $p \geq 1$ guarantees the accuracy of expectations for any function f with small Lipschitz constant with respect the metric $d_\Sigma(\theta, \theta') := \|\Sigma^{-1/2}(\theta - \theta')\|_2$; that is, when $\sup_{\theta \neq \theta'} |f(\theta) - f(\theta')|/d_\Sigma(\theta, \theta')$ is small.

While the Wasserstein distance controls the error in mean and covariance estimates, it does not provide strong control on interval probability estimates. The KL divergence, however, does, since for distributions η and ζ, $|I_{\eta,i,a,b} - I_{\zeta,i,a,b}| \leq \sqrt{\mathrm{KL}(\eta \mid \zeta)/2}$ for all $a < b$ and i (see Appendix B.2). As we show next, in many scenarios we can bound the Wasserstein distance by the KL divergence and therefore enjoy the benefits of both. Our result is based on the following definition, which makes the notion of the scale of a distribution precise:

Definition 4 *For $p \geq 1$ and a positive-definite matrix $\Sigma \in \mathbb{R}^{d \times d}$, the distribution η is said to be* (p, Σ)-exponentially controlled *if*

$$\inf_{\theta'} \log \int e^{\|\Sigma^{-1/2}(\theta - \theta')\|_2^p} \eta(\mathrm{d}\theta) \leq d/2. \tag{4}$$

Specially, $\Sigma^{1/2}$ establishes the appropriate scale for uncertainty with respect to η. For example, if $\eta = \mathcal{N}(m, V)$, then it is a straightforward exercise to confirm that η is $(2, 1.78^2 V)$-exponentially controlled.

The following result establishes the relevant link between the KL divergence and the Wasserstein distance via Definition 4.

Proposition 5 *If η is (p, Σ)-exponentially controlled, then for all ζ absolutely continuous with respect to η,*

$$\mathcal{W}_{p,\Sigma}(\zeta, \eta) \leq (3 + d)\mathcal{K}_p(\zeta \mid \eta),$$

where $\mathcal{K}_p(\zeta \mid \eta) := \mathrm{KL}(\zeta \mid \eta)^{\frac{1}{p}} + \{\mathrm{KL}(\zeta \mid \eta)/2\}^{\frac{1}{2p}}$.

Proof The result follows from Bolley and Villani (2005, Corollary 2.3) after the change-of-variable $\theta \mapsto \Sigma^{-1/2}\theta$ and using the fact that the KL divergence is invariant under diffeomorphisms, then applying Eq. (4). ■

If ζ and η could operate over different scales, then we can use the *symmetrized KL divergence* $\mathrm{SKL}(\zeta, \eta) := \mathrm{KL}(\zeta \mid \eta) + \mathrm{KL}(\eta \mid \zeta)$. Indeed, it follows from Proposition 5 that if $\mathrm{SKL}(\zeta, \eta)$ is small, then the (p, Σ)-Wasserstein distance is small whenever *either* η or ζ is (p, Σ)-exponentially controlled.

4.2 An Idealized Termination Rule

Based on our developments in Section 4.1, we will define our termination rule in terms of the symmetrized KL divergence. Recall that q_* denotes the optimal variational approximation to π and $\hat{q}_*$ denotes an estimate of q_*. Since the total variation and $(1, \Sigma)$-Wasserstein distances are controlled by the square root of the KL divergence, we focused on the square root of the symmetrized KL divergence. For the current learning rate $\gamma > 0$, let $q_{\gamma *} := q_{\bar{\lambda}_\gamma}$ denote the target γ–learning-rate variational approximation. The termination rule we propose is based on the trade-off between the improved accuracy of the approximation if the learning rate were reduced to $\rho\gamma$ and the time required to reach that improved accuracy.

To quantify the improved accuracy, we introduce a user-chosen target accuracy target ξ for $\mathrm{SKL}(q_*, \hat{q}_*)^{1/2}$. If the user expects $\mathrm{KL}(\pi \mid q_*)$ to be large, then setting ξ to a moderate value such as 1 or 10 could give acceptable performance. If the user expects $\mathrm{KL}(\pi \mid q_*)$ to be small, then setting ξ to a value such as 0.1 or 0.01 might be more appropriate. Using ξ, define the *relative SKL improvement*

$$\mathrm{RSKL} := \frac{\mathrm{SKL}(q_*, q_{\rho\gamma *})^{1/2} + \xi}{\mathrm{SKL}(q_*, q_{\gamma *})^{1/2}},$$

where the first term measures the relative improvement of the approximation if the learning rate were reduced and the second term measures the current accuracy relative to the desired accuracy. To quantify the time to obtain the relative accuracy improvement, we use the number of iterations to reach convergence for the fixed learning rate $\rho\gamma$. Letting $K_{\gamma *}$ denote the number of iterations required to reach the target γ–learning-rate variational approximation, we define the *relative iteration increase*

$$\mathrm{RI} := \frac{K_{\rho\gamma *}}{K_{\gamma *} + K_0},$$

where K_0 denotes the number of iterations the user would consider "small". Combining RSKL and RI, we obtain the *inefficiency index* $\mathcal{I} = \mathrm{RSKL} \times \mathrm{RI}$, the relative improvement in accuracy times the relative increase in runtime. Thus, we can interpret $\mathcal{I}$ as quantifying how much greater the increase in runtime cost (above a baseline of K_0 iterations) will be compared to the reduction in error (down to a target error of ξ). For example, $\mathcal{I} = 2$ means the increase in runtime cost is twice as large as the reduction in error. Our *idealized SKL inefficiency termination rule* triggers when $\mathcal{I} > \tau$, where τ is a user-specified inefficiency threshold that allows the user to trade off accuracy with computation, but only up to the point where $\mathrm{SKL}(q_*, q_{\gamma *})^{1/2} \approx \xi$.

4.3 An Implementable Termination Rule

The idealized SKL inefficiency termination rule cannot be directly implemented since q_* is unknown; and if it were known, it would be unnecessary to run the optimization algorithm. However, we will show that it is possible to obtain a good estimate of the symmetrized KL divergence between the approximation obtained with a given learning rate γ' and the optimal approximation without access to q_*.

Recall that $q_{\gamma *}$ denotes the target γ–learning-rate variational approximation. With a slight abuse of notation, we let the optimal zero–learning-rate approximation refer to the optimal approximation: $q_{0*} := \lim_{\gamma \to 0} q_{\gamma *} = q_*$. Our approach is motivated by Theorem 1 and in particular the form of the bias $\bar{\lambda}_\gamma - \lambda_*$ in Eq. (2). We first consider the still-common setting when $\mathcal{Q}$ is the family of mean-field Gaussian distributions, where the parameter $\lambda = (\tau, \psi) \in \mathbb{R}^{2d}$ corresponds to the distribution $q_\lambda = \mathcal{N}(\tau, \mathrm{diag}\, e^{2\psi})$.

Proposition 6 *Let $\mathcal{Q}$ be the family of mean-field Gaussian distributions. If Eq. (2) holds and $\gamma' = O(\gamma)$, then there is a constant $C > 0$ depending only on A and λ_* such that*

$$\mathrm{SKL}(q_{\gamma *}, q_{\gamma' *}) = C(\gamma - \gamma')^2 + o(\gamma^2).$$

See Appendix A.1 for the proof. Assuming that the current learning rate is γ, then the previous learning rate was γ/ρ. Let $\delta_\gamma := \mathrm{SKL}(q_{\gamma*}, q_{\gamma/\rho*})$ denote the symmetrized KL divergence between the optimal variational approximations obtained at each of these learning rates. In principle we can use Proposition 6 to estimate C by

$$\hat{C} = \delta_\gamma \rho^2 / \{\gamma^2 (1-\rho)^2\},$$

and then estimate that

$$\mathrm{SKL}(q_{\gamma*}, q_*) \approx \hat{C}\gamma^2 = \delta_\gamma \rho^2 / (1-\rho)^2.$$

There are, however, two problems with the tentative approach just outlined. The first problem is that Theorem 1 only holds for standard SGD; however, adaptive SGD algorithms are widely used in practice. Indeed, we observe empirically that $\mathrm{SKL}(q_{\gamma*}, q_{\gamma'*}) = \Theta(|\gamma - \gamma'|^{\kappa/2})$ with $\kappa \approx 1$ for RMSProp (Fig. C.6) and $\kappa \in (1, 1.6)$ (with a point estimate at 1.2) for Adam Fig. C.7). Hence RMSProp and Adam both appear to have larger errors than SGD when the step size is small. To get the accuracy of SGD but also adaptivity, we modify the adaptive gradient methods to behave asymptotically (in the number of iterations) like SGD. In the cases of RMSProp and Adam, we propose *averaged RMSProp* (avgRMSProp) and *averaged Adam* (avgAdam), which use the squared gradient update

$$\nu^{(k+1)} = \beta_k \nu^{(k)} + (1-\beta_k)\hat{g}^{(k)} \odot \hat{g}^{(k)},$$

for $\beta_k = 1 - 1/k$. Hence, $\nu^{(k+1)} = (k+1)^{-1} \sum_{k'=0}^{k} \hat{g}^{(k)} \odot \hat{g}^{(k)}$ is the averaged squared gradient over all iterations (Mukkamala and Hein, 2017, §4). As long as the SGD Markov chain is ergodic and $\mathbb{E}[\|\hat{g}^{(k)}\|_2^2] < \infty$ at stationarity, $\nu^{(k)}$ converges almost surely to a constant and hence the SGD bias analysis also applies to avgRMSProp and avgAdam.

The second problem is that Proposition 6 only holds for the mean-field Gaussian variational family. However, other variational families such as normalizing flows are of substantial practical interest. Therefore, we consider the weaker assumption that either Eq. (2) holds *or* there exist constant vectors $\Lambda, A \in \mathbb{R}^m$ and a constant $\kappa \in [1/2, 1)$ such that

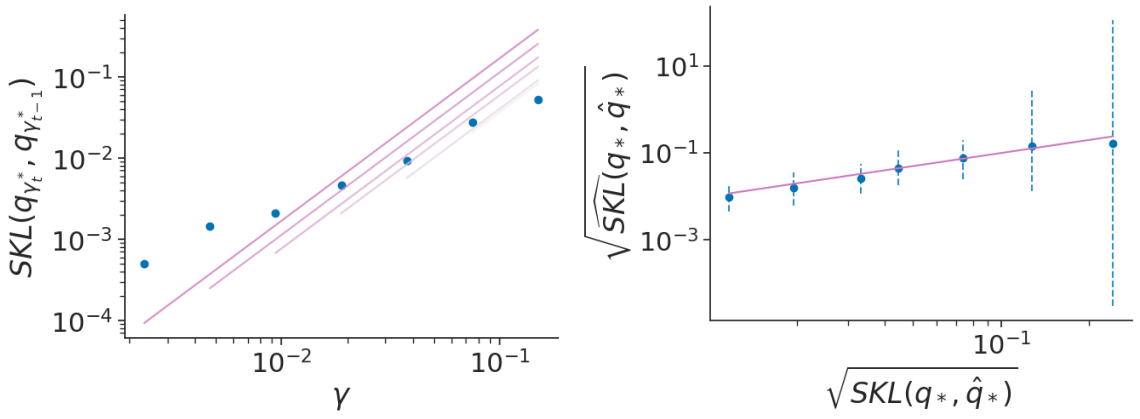
$$\bar{\lambda}_\gamma = \lambda_* + \Lambda\gamma^\kappa + A\gamma + o(\gamma^{2\kappa}). \tag{5}$$

Adding the latter assumption, we have the following generalization of Proposition 6, which holds for any sufficiently regular variational family:

Proposition 7 *Let $\mathcal{Q}$ be the variational approximation family. If (i) Eq. (5) holds for some $\kappa \in [1/2, 1)$ or Eq. (2) holds (in which case let $\kappa = 1$), (ii) $\gamma' = O(\gamma)$, and (iii) for all $\theta \in \mathbb{R}^d$, $\log q_\lambda(\theta)$ is three-times continuously differentiable with respect to λ, then for some $C \geq 0$,*

$$\mathrm{SKL}(q_{\gamma*}, q_{\gamma'*}) = C\{\gamma^\kappa - (\gamma')^\kappa\}^2 + o(\gamma^{2\kappa}).$$

Moreover, C depends on only λ_* and either Λ (if $\kappa \in [1/2, 1)$) or A (if $\kappa = 1$).



(a) (b)

Figure 3: Results for estimating the symmetrized KL divergence with avgAdam in the case of a Gaussian distribution $\mathcal{N}(0, V)$ with $d = 100$ and $V_{ij} = j\mathbb{1}[i = j]$ (diagonal non-identity covariance). **(a)** Learning rate versus symmetrized KL divergence of adjacent iterate averaged estimates of the optimal variational distribution. The lines indicate the linear regression fits, with setting $\kappa = 1$. **(b)** Square root of true symmetrized KL divergence versus the estimated value with 95% credible interval. The uncertainty of the estimates decreases and remains well-calibrated as the learning rate decreases.

See Appendix A.2 for the proof. Using Proposition 7 and omitting $o(\gamma^{2\kappa})$ terms, we have $\delta_\gamma \approx C\gamma^{2\kappa}(1/\rho^\kappa - 1)^2$. To improve the reliability of the estimates based on Proposition 7, we propose to use the symmetrized KL estimates between the variational approximations obtained at successive fixed learning rates. Let γ_0 denote the initial learning rate, so that after t learning rate decreases, the learning rate is $\gamma_t := \gamma_0\rho^t$. Let $\delta_t := \mathrm{SKL}(q_{\gamma_t *}, q_{\gamma_{t-1} *})$ and assume the current learning rate is γ_T.

Depending on the optimization algorithm, we can estimate κ (or set $\kappa = 1$ if using modified adaptive SGD algorithm with a mean-field Gaussian variation family) and C using a regression model of the form

$$\log \delta_t = \log C + 2\log(1/\rho^\kappa - 1) + 2\kappa \log \gamma_t + \eta_t, \qquad t = 1, \ldots, T, \tag{6}$$

where $\eta_t \sim \mathcal{N}(0, \sigma^2)$. Given the estimate $\hat{C}$ for C and the estimate $\hat{\kappa}$ for κ (or $\hat{\kappa} = 1$) , we obtain the estimated relative SKL,

$$\widehat{\mathrm{RSKL}} = \rho^{\hat{\kappa}} + \frac{\xi}{\hat{C}^{1/2}\gamma_t^{\hat{\kappa}}}.$$

Because we use the regression model in Eq. (6) in a low-data setting, we place (weak) priors on $\log C$ and σ:

$$\log C \sim \mathsf{Cauchy}(0, 10), \qquad \sigma \sim \mathsf{Cauchy}^+(0, 10),$$

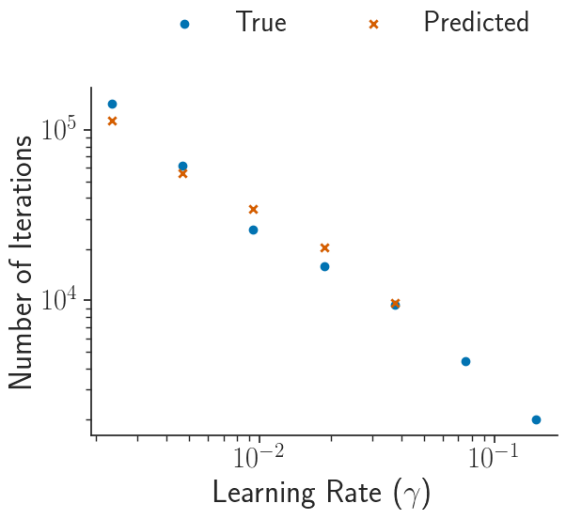


Figure 4: Results for predicting the number of iterations needed to reach convergence at each learning rate decrease in the case of Gaussian distribution $\mathcal{N}(0, V)$ with $d = 100$ and $V_{ij} = j\mathbb{1}[i = j]$ (diagonal non-identity covariance). The blue points (orange crosses) represent the true (predicted) number of iterations needed to reach convergence.

where Cauchy^+ is the Cauchy distribution truncated to nonnegative values. If we use an adaptive stochastic optimization algorithm then we also place a prior on κ:

$$\kappa \sim \mathsf{Unif}(0, 1).$$

Also, because we expect early SKL estimates to be less informative about C (and κ) due to the influence of $o(\gamma^{2\kappa})$ terms, we use a weighted regression with the likelihood term for (δ_t, γ_t) having weight

$$w_t = \{1 + (T - t)^2/3^2\}^{-1/4}. \tag{7}$$

The weight formula enables the amplification of the significance of the most recent observations, with down-weighting becomes more significant after there are about 3 additional observations. On the other hand, the power of $1/4$ ensures a gradual reduction in weight, preventing a steep drop-off in importance.

We use the posterior mean(s) to estimate $\hat{C}$ (and $\hat{\kappa}$). Figure 3 validates that, in the case of avgAdam, the log of the learning rate and symmetrized KL divergence have approximately a linear relationship and that our regression approach to estimating C leads to reasonable estimates of $\mathrm{SKL}(q_{\gamma*}, q_*)$. See Fig. C.2 for similar results for other target distributions with avgAdam.

To estimate the relative iteration increase RI, we need to estimate the number of iterations to reach convergence at the next learning rate γ_{t+1}. It is reasonable to assume

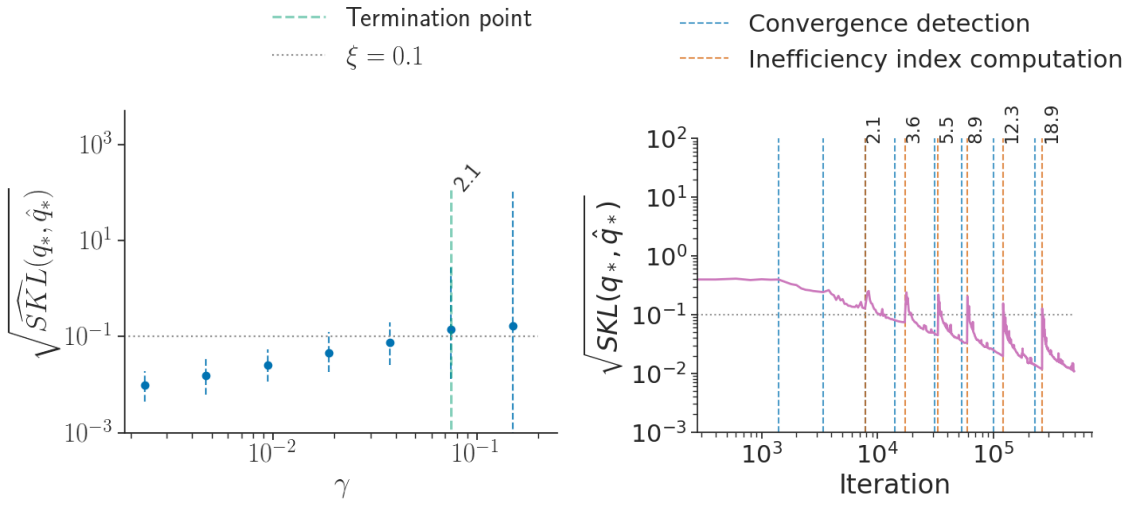


Figure 5: Results for the termination rule trigger point in the case of a Gaussian distribution $\mathcal{N}(0, V)$ with $d = 100$ and $V_{ij} = j\mathbb{1}[i = j]$ (diagonal non-identity covariance). **(a)** Learning rate versus square root of estimated symmetrized KL divergence with 95% credible interval (dashed blue line). The green vertical line indicates the termination rule trigger point with the corresponding $\hat{\mathcal{I}}$ value. **(b)** Iterations versus square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the convergence detection points using $\hat{R}$ (blue) and inefficiency index computation ($\hat{\mathcal{I}}$) points (orange) with corresponding values.

that there is exponential growth in the number of iterations to reach convergence as the learning rate decreases since stochastic gradient algorithms to converge at a polynomial rate (Bubeck, 2015). Recall that K_{γ_t} is the number of iterations to reach convergence at the current learning rate. We fit a weighted least square regression model of the form

$$\log K_{\gamma_t} = \alpha \log \gamma_t + \beta + \nu_t, \qquad t = 1, \ldots, T, \tag{8}$$

where $\nu_t \sim \mathcal{N}(0, \sigma_t^2)$. We then use the coefficient estimates $\hat{\alpha}$ and $\hat{\beta}$ to predict the number of iterations required for convergence at the next learning rate to be $\widehat{K}_{\gamma_{t+1}} := \gamma_{t+1}^{\hat{\alpha}} e^{\hat{\beta}}$. We use the same weights given in Eq. (7) for observations of the regression model due to the non-linear behavior of the earlier convergence iterate estimates. Figure 4 demonstrates that linear relationship in Eq. (8) does in fact hold and that our weighted least square regression model predicts the number of convergence iterations K_{γ_t} quite accurately. The estimated relative iterations is then $\widehat{\mathrm{RI}} = \widehat{K}_{\gamma_{t+1}}/(K_{\gamma_t} + K_0)$.

Using the estimates $\widehat{\mathrm{RSKL}}$ and $\widehat{\mathrm{RI}}$ we obtain the termination rule $\widehat{\mathcal{I}} = \widehat{\mathrm{RSKL}} \times \widehat{\mathrm{RI}} > \tau$. Figure 5 shows that when the user chosen target accuracy $\xi = 0.1$, the termination rule triggers when the square root of the symmetrized KL divergence is approximately equal to ξ. Figures C.3 to C.5 shows similar results of other Gaussian targets and `posteriordb` models and data sets (see Section 7.2 for details).

5. Learning Rate Scheduler

For a fixed learning rate, computing the iterate average $\hat{\lambda}_\gamma$ defined in Eq. (3) requires determining the iteration k^{conv} at which stationarity is reached and the number of iterations k^{avg} to use for computing the average. We address each of these in turn.

5.1 Detecting Convergence to Stationarity

We investigate two approaches to detecting stationarity: the SASA+ algorithm of Zhang et al. (2020) and the $\widehat{R}$-based criterion from Dhaka et al. (2020). We make several adjustments to both approaches to reduce the number of tuning parameters and to make the remaining ones more intuitive. In our empirical findings, we have observed that the $\widehat{R}$ criterion outperforms the SASA+ criterion. Therefore, we describe the former here and the latter in Appendix B.3.

Let $\widehat{R}(\lambda_i^{(k-W+1)}, \ldots, \lambda_i^{(k)})$ denotes the split-$\widehat{R}$ of the ith component of the last W iterates and define

$$\widehat{R}_{\max}(W) := \max_{1 \le i \le m} \widehat{R}(\lambda_i^{(k-W+1)}, \ldots, \lambda_i^{(k)}).$$

An $\widehat{R}_{\max}(W)$ value close to 1 indicates the last W iterates are close to stationarity. In MCMC applications having $\widehat{R}_{\max}(W) \le 1.01$ is desirable (Vats and Knudson, 2021; Vehtari et al., 2021). Dhaka et al. (2020) uses the weaker condition $\widehat{R}_{\max}(W) \le 1.1$ since iterate averaging does not require the same level of precision as MCMC. Dhaka et al. (2020) take the window size $W = 100$, but in more challenging and high-dimensional problems a fixed smaller W is insufficient. Therefore, we instead search over window sizes between a minimum window size $W_{\min}$ and $0.95k$ to find the one that minimizes $\widehat{R}_{\max}(W)$. The

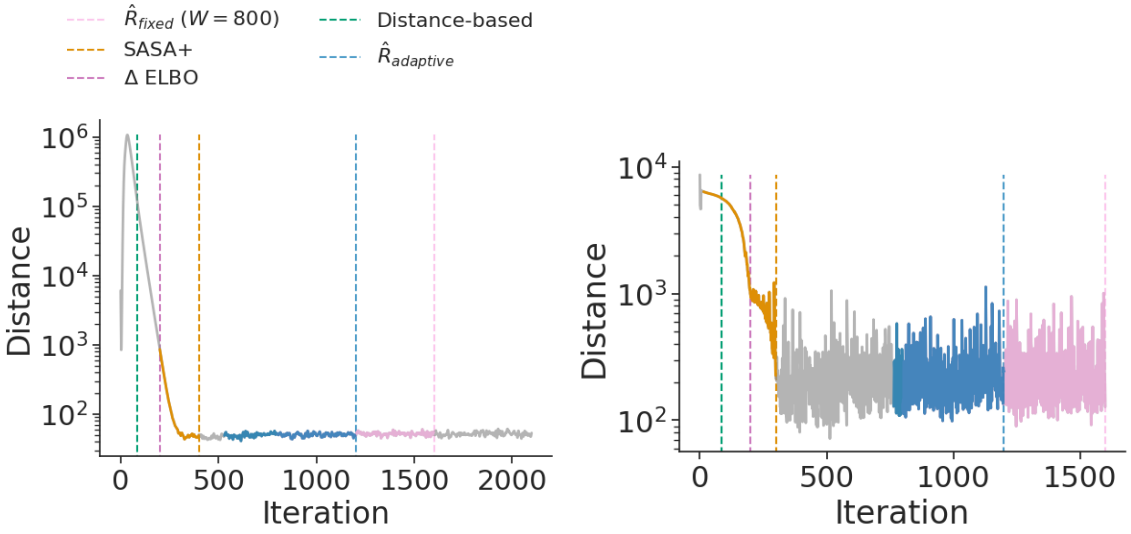


Figure 6: Iteration number versus distance between iterate average and current iterate. The vertical lines indicate convergence detection trigger points and (for SASA+ and $\widehat{R}$) the colored portion of the accuracy values indicate they are part of the window used for convergence detection. **(a)** An uncorrelated Gaussian distribution $\mathcal{N}(0, V)$ with $d = 500$ and $V = I$. **(b)** A `posteriordb` data set/model *mcycle_gp* with $d = 66$.

minimum window size is necessary to ensure the $\widehat{R}$ values are reliable. We use the upper bound $0.95k$ to always allow a small amount of "warm-up" without sacrificing more than 5% efficiency. Therefore, we estimate $W_{\text{opt}} = \arg\min_{W_{\min} \leq W \leq 0.95k} \widehat{R}_{\max}(W)$ using a grid search over 5 equally spaced values ranging from $W_{\min}$ to $0.95k$ and require $\widehat{R}_{\max}(W_{\text{opt}}) \leq 1.1$ as the stationarity condition.

Figure 6 compares our adaptive SASA+ and adaptive $\widehat{R}$ criteria to the criterion used in Dhaka et al. (2020) with a fixed window size of $W = 800$ and ΔELBO rule from Kucukelbir et al. (2015), which is used Stan's ADVI implementation (cf. the results of Dhaka et al. (2020)). We do not use $W = 100$ as is done by Dhaka et al. (2020) because it was too small to detect convergence. Additionally, Fig. 6 compares to another convergence detection approach proposed by Pesme et al. (2020) (described in Appendix B.4), where they use a distance-based statistic to detect convergence. While adaptive SASA+, ΔELBO, fixed window size $\widehat{R}$, and the distance-based statistic approach sometimes trigger too early or too late or SASA+ use iterations before it reaches the convergence, adaptive $\widehat{R}$ consistently triggers when the full window suggests convergence has been reached. See Fig. C.1 for additional Gaussian target examples.

5.2 Determining the Number of Iterates for Averaging

After detecting convergence to stationarity, we need to find k^{avg} large enough to ensure the iterative average is sufficiently close to the mean $\bar{\lambda}_{\gamma t}$. But what is close enough? Building on our discussion in Section 3, we aim to ensure the error in the variational parameter estimates are small relative to the scale of uncertainty. For mean-field Gaussian distributions, the following result allows us to make such a guarantee precise.

Proposition 8 *Let $\mathcal{Q}$ be the family of mean-field Gaussian distributions. Let $\hat{\lambda} = (\hat{\tau}, \hat{\psi})$ denote an approximation to $\bar{\lambda} = (\bar{\tau}, \bar{\psi})$. Define $\hat{\sigma} := \exp(\hat{\psi})$ and $\bar{\sigma} := \exp(\bar{\psi})$. If there exists $\varepsilon \in (0, 1/2)$ such that $|\hat{\tau}_i - \bar{\tau}_i| \leq \varepsilon \hat{\sigma}_i$ and $|\hat{\psi}_i - \bar{\psi}_i| \leq \varepsilon$, then*

$$\frac{|\hat{\sigma}_i - \bar{\sigma}_i|}{\bar{\sigma}_i} \leq 1.5\varepsilon \qquad \textit{and} \qquad \frac{|\hat{\tau}_i - \bar{\tau}_i|}{\bar{\sigma}_i} \leq 1.75\varepsilon.$$

See Appendix A.4 for the proof. Based on Proposition 8, for mean-field Gaussian variational families we use the iterate average once the mean MCSEs $d^{-1}\sum_{i=1}^{d} \text{MCSE}(\hat{\tau}_{\gamma,i})/\hat{\sigma}_{\gamma,i}$ and $d^{-1}\sum_{i=1}^{d} \text{MCSE}(\hat{\psi}_{\gamma,i})$ are less than ε.For other variational families we rely on the less rigorous condition that $m^{-1}\sum_{i=1}^{m} \text{MCSE}(\hat{\lambda}_{\gamma,i})$ is less than ε. We also require the effective sample sizes of all parameters to be at least 50 to ensure the MCSE estimates are reliable.

Because the MCSE check requires computing d ESS values, it can be computationally expensive, especially for high-dimensional models. Therefore, it is important to optimize when conducting the checks.A well-known approach in such situations is the "doubling trick." Let W_{conv} denote the window size when convergence is detected, and let W_{opt} denote the minimal window size that satisfies the MCSE check. The doubling trick would suggest checking at iteration numbers $k^{\text{conv}} + 2^j W_{\text{conv}}$ for $j = 0, 1, \ldots,$ in which case the total computational cost is within a factor of $4 \log W_{\text{opt}}$ of the optimal scenario in in which the check is only done at $k^{\text{conv}} + W_{\text{conv}}$ and $k^{\text{conv}} + W_{\text{opt}}$). However, we can potentially do substantially better by accounting for the different computational cost of the optimization versus the MCSE check.

Proposition 9 *Assume that the cost of the MCSE check using K iterates is $C_E K$ and the cost of K iterations of optimization is $C_O K$. Let $r := C_O/C_E$, $\chi(r) := 1 + (1+r)^{-1/2}$, and $g(r) := (2 + r + 2(1+r)^{1/2})/(1+r)$. If the MCSE check is done on iteration numbers $k^{\text{conv}} + \chi(r)^j W_{\text{conv}}$ for $j = 0, 1, \ldots,$ then the total computational cost will be within a factor of $g(r)$ of optimal.*

See Appendix A.5 for the proof. Since $g(0) = 4$ and $g(r)$ is monotonically decreasing in r, when $r \approx 0$; that is, C_O is negligible compared to C_E, we recover the doubling rule since $\chi(0) = 2$. However, as long as r is significantly greater than zero, the worst-case additional cost factor can be substantially less than 4. Therefore we carry out the MCSE check on iteration numbers $k^{\text{conv}} + \chi(r)^j W_{\text{conv}}$ with r estimated based on the actual runtimes of the optimization so far and the first MCSE check.

6. Complete Framework

Combining our innovations from Sections 4 and 5 leads to our complete framework. When γ is fixed, our proposal from Section 5 is summarized in Algorithm 1, which we call *fixed–learning-rate automated stochastic optimization* (FASO). Combining the termination rule

from Section 4 with FASO, we get our complete framework, *robust and automated black-box variational inference (RABVI)*, which we summarize in Algorithm 2. We will verify the robustness of RABVI through numerical experiments. RABVI is automatic since the user is only required to provide a target distribution and the only tuning parameters we recommend changing from their defaults are defined on interpretable, intuitive scales:

- **accuracy threshold ξ:** The symmetrized KL divergence accuracy threshold can be set based on the expected accuracy of the variational approximation. If the user expects $\mathrm{KL}(\pi \mid q_*)$ to be large, then we recommend choosing $\xi \in [1, 10]$. If the user expects $\mathrm{KL}(\pi \mid q_*)$ to be fairly small, then we recommend choosing $\xi \in [.01, 1]$. Our experiments suggest $\xi = 0.1$ is a good default value.

- **inefficiency threshold τ:** We recommend setting the inefficiency threshold $\tau = 1$, as this weights accuracy and computation equally. A larger value (for example, 2) could be chosen if accuracy is more important while a smaller value (for example, $1/2$) would be appropriate if computation is more of a concern.

- **maximum number of iterations $K_{\max}$:** The maximum number of iterations can be set by the user based on their computational budget. RABVI will warn the user if the maximum number of iterations is reached without convergence, so the user can either increase $K_{\max}$ or accept the estimated level of accuracy that has been reached.

We expect the remaining tuning parameters will typically not be adjusted by the user. We summarize our recommendations:

- **initial learning rate γ_0:** When using adaptive methods such as RMSProp or Adam, the initial learning rate can essentially be set in a problem-independent manner. We use $\gamma_0 = 0.3$ in all of our experiments. If using non-adaptive methods, a line search rule such as the one proposed in Zhang et al. (2020) could be used to find a good initial learning rate.

- **minimum window size $W_{\min}$:** We recommend taking $W_{\min} = 200$ so that that each of the split-$\widehat{R}$ values are based on at least 100 samples.

- **small iteration number K_0:** The value of K_0 should represent a number of iterations the user considers to be fairly small (that is, not requiring too much computational effort). We use $K_0 = 5W_{\min} = 1000$ for our experiments, but it could be adjusted by the user.

- **initial iterate average relative error threshold ε_0:** We recommend scaling ε_0 with ξ since more accurate iterate averages are required for sufficiently accurate symmetrized KL estimates. Therefore, we take $\varepsilon_0 = \xi$ by default.

- **adaptation factor ρ:** We recommend taking $\rho = 0.5$ because using a smaller ρ value could lead to too few δ_t values for the estimation of C (and κ) and using a larger ρ value would make the algorithm too slow.

- **Monte Carlo samples M:** We find that $M = 10$ provides a good balance between gradient accuracy and computational burden but the performance is fairly robust to the choice of M as long as it is not too small.

Algorithm 1: Fixed–learning-rate automated stochastic optimization (FASO)

Input: initial variational parameter $\lambda^{(0)}$,
learning rate γ,
minimum window size $W_{\min}$,
initial iterate average relative error threshold ε,
maximum iterations $K_{\max}$

$k^{\text{conv}} \leftarrow$ **null** `// iteration when stationarity reached`
success $\leftarrow$ **false**
for $k = 1, \ldots, K_{\max}$ **do**
  compute stochastic gradient $\hat{g}^{(k)}$
  compute descent direction d_k
  `// step in descent direction`
  $\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \gamma d_k$
  if $k^{\text{conv}} =$ **null** *and* $k \mod k_{\text{check}} = 0$ **then** check for convergence
    `// define window-based ESS`
    $\widehat{R}_{\max}(W) := \max_{1 \le i \le m} \widehat{R}(\lambda_i^{(k-W+1)}, \ldots, \lambda_i^{(k)})$
    `// compute optimal window`
    $W_{\text{opt}} \leftarrow \arg\min_{W_{\min} \le W \le 0.95k} \widehat{R}_{\max}(W)$
    if $\widehat{R}_{\max}(W_{\text{opt}}) \le 1.1$ **then**
      $k^{\text{conv}} \leftarrow k - W_{\text{opt}}$
      `// window size at which to check MCSE`
      $W_{\text{check}} \leftarrow W_{\text{opt}}$
      $\chi_* \leftarrow \chi(r)$ `// see Prop. 9`
  if $k^{\text{conv}} \neq$ **null** *and* $k - k^{\text{conv}} = W_{\text{check}}$ **then** check for accuracy of iterate average
    $W \leftarrow W_{\text{check}}$
    $\hat{\lambda} \leftarrow W^{-1} \sum_{i=k-W+1}^{k} \lambda^{(i)}$
    $e \leftarrow \text{MCSE}(\lambda^{(k-W)}, \ldots, \lambda^{(k)})$
    $\text{ESS}_{\min} \leftarrow \min_i \text{ESS}(\lambda_i^{(k-W)}, \ldots, \lambda_i^{(k)})$
    if *mean-field Gaussian family* **then** $e_i \leftarrow e_i / \exp(\hat{\psi}_i)$ for $i = 1, \ldots, d$
    if mean $e_i < \varepsilon$ *and* $\text{ESS}_{\min} \ge 50$ **then**
      success $\leftarrow$ **true**
      break
    else
      $W_{\text{check}} \leftarrow \chi_* W$
$\bar{\lambda} \leftarrow W^{-1} \sum_{i=k-W+1}^{k} \lambda^{(i)}$
return $(k, \bar{\lambda}, \textit{success})$

Algorithm 2: Robust and automated black-box variational inference (RABVI)

Input: initial variational parameter $\lambda^{(0)}$,
maximum number of iterations $K_{\max}$,
initial learning rate γ_0 (default: 0.3),
minimum window size $W_{\min}$ (default: 200),
accuracy threshold ξ (default: 0.1),
inefficiency threshold τ (default: 1.0),
initial iterate average error threshold ε_0 (default: 0.1),
adaptation factor ρ (default: 0.5)
small iteration number K_0 (default: 1000)

$\bar{\lambda}_{\text{curr}} \leftarrow \lambda^{(0)}$ `// current iterate average`
$\gamma \leftarrow \gamma_0$ `// learning rate`
$\varepsilon \leftarrow \varepsilon_0$ `// iterate average error threshold`
$k \leftarrow 0$ `// total iterations`
$t \leftarrow 0$ `// total epochs`
while $k < K_{\max}$ **do**
  $\bar{\lambda}_{\text{prev}} \leftarrow \bar{\lambda}_{\text{curr}}$ `// record previous iterate average`
  $k_{\text{new}}, \bar{\lambda}_{\text{curr}}, \mathit{success} \leftarrow$ `FASO(`$\bar{\lambda}_{\text{curr}}, \gamma, W_{\min}, \varepsilon, K_{\max} - k$`)`
  if not *success* **then**
    print "Warning: failed to converge. Estimated error is *error*"
    break
  $k \leftarrow k + k_{\text{new}}$ `// update total iterations`
  if $t \geq 1$ **then**
    $\delta_t \leftarrow \mathrm{SKL}(q_{\bar{\lambda}_{\text{prev}}}, q_{\bar{\lambda}_{\text{curr}}})$
    compute estimates $\hat{C}$ and $\hat{\kappa}$ using weighted linear regression
    $\widehat{\mathrm{RSKL}} \leftarrow \rho^{\hat{\kappa}} + \xi/(\hat{C}^{1/2}\gamma^{\hat{\kappa}})$
    $\hat{K}_\gamma \leftarrow k_{\text{new}}^{(t)} - k_{\text{new}}^{(t-1)}$
    if $t \geq 2$ **then**
      `// remove the converged iterations of initial variation parameter`
      compute estimates $\hat{\alpha}$ and $\hat{\beta}$ using weighted least squares
      if $\hat{\beta} < 0$ **then**
        $\hat{K}_{\rho\gamma} \leftarrow (\rho\gamma)^{\hat{\alpha}} e^{\hat{\beta}}$
      else
        $\hat{K}_{\rho\gamma} \leftarrow \hat{K}_\gamma$
      $\widehat{\mathrm{RI}} \leftarrow \hat{K}_{\rho\gamma}/(\hat{K}_\gamma + K_0)$
      if $\widehat{\mathrm{RSKL}} \cdot \widehat{\mathrm{RI}} > \tau$ **then**
        break
  $\gamma \leftarrow \rho\gamma$ `// decrease learning rate`
  $\varepsilon \leftarrow \rho\varepsilon$ `// decrease iterate average error threshold`
  $t \leftarrow t + 1$ `// increment epoch counter`
return $\bar{\lambda}_{\text{curr}}$

7. Experiments

Unless stated otherwise, all experiments use avgAdam to compute the descent direction, mean-field Gaussian distributions as the variational family, and the tuning parameters values recommended in Section 6. We fit the regression model for C (and κ) in Stan, which result in extremely small computational overhead of less than 0.5%. We compare RABVI to FASO, Stan's ADVI implementation, SGD using an exponential decay of the learning rate, and fixed–learning rate versions of RMSProp, Adam, and a windowed version of Adagrad (WAdagrad), which is the default optimizer in PyMC3. Moreover, we compare RABVI with exponential decay and cosine learning rate schedules using Adam and RMSProp optimization methods. We run all the algorithms that do not have a termination criterion for $K_{\max} = 100{,}000$ iterations and for the fixed–learning-rate algorithms we use learning rate $\gamma = 0.01$ in an effort to balance speed with accuracy. For exponential decay, we use a learning rate of $\gamma = \gamma_0 \delta^{\lfloor k/s \rfloor}$, where γ_0 denotes the initial learning rate, δ denotes the decay rate, k denotes the iteration, and s denotes the decay step. We choose $\gamma_0 = 0.01$, $\delta = 0.96$, and $s = 900$ so that the final learning rate is approximately 0.0001 (Chen et al., 2017). For cosine schedule, we use a learning rate of $\gamma = \gamma_{\min} + \frac{1}{2}(\gamma_{\max} - \gamma_{\min})(1 + \cos(\frac{k}{K}\pi))$, where $\gamma_{\min}$ and $\gamma_{\max}$ denote the minimum and maximum values of learning rate, k denotes the current iteration, and K denotes the maximum number of iterations (Loshchilov and Hutter, 2017). We choose $\gamma_{min} = 0.0001$, $\gamma_{max} = 0.01$ to make it comparable with other methods.

We use symmetrized KL divergence as the accuracy measure when we can compute the ground-truth optimal variational approximation. Otherwise, we use the following metrics (where μ and σ are, respectively, the posterior mean and standard deviation vectors):

- *Relative mean error* $\|(\mu - \hat{\mu})/\sigma\|_2$, where $\hat{\mu}$ is the variational approximation to μ.
- *Relative standard deviation error* $\|\hat{\sigma}/\sigma - 1\|_2$, where $\hat{\sigma}$ is the variational approximation to σ.
- *Under coverage error* of the variational approximation to the 95% credible intervals $\min(0, |.95 - c_i|)$, where $c_i := \Pi(\{\theta : \theta_i \in (a_i, b_i)\})$ and (a_i, b_i) is the variational estimate of the central 95% credible interval for parameter θ_i.
- *Maximum mean discrepancy (MMD)* $\mathrm{MMD}^2(P, Q) := \mathbb{E}[k(x, x') - 2\,k(x, y) + k(y, y')]$, where $x, x' \sim P$ and $y, y' \sim Q$ are independent and $k(x, y) = \exp\{-\frac{1}{2}\left\|\frac{x-y}{\sigma}\right\|_2^2\}$ is the squared exponential kernel (Gretton et al., 2006).

7.1 Accuracy with Gaussian Targets

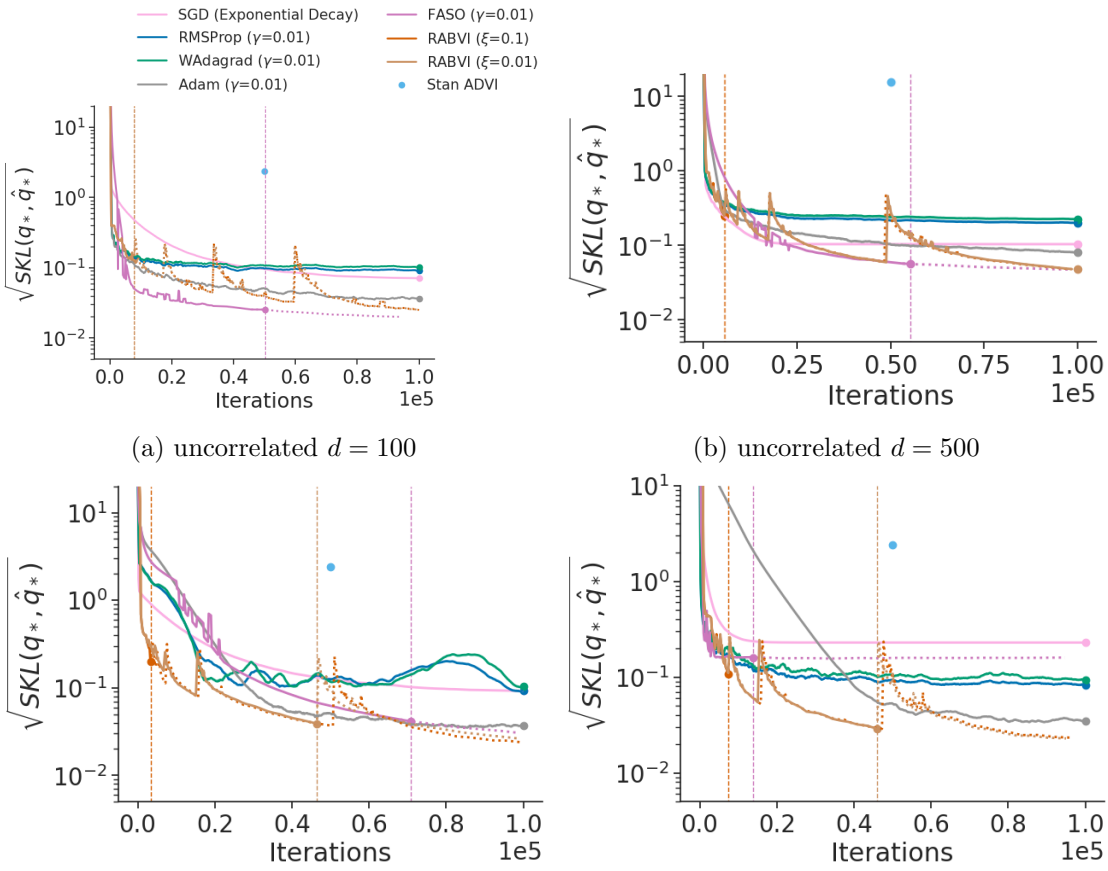
First, to explore optimization accuracy relative to the optimal variational approximation, we consider Gaussian targets of the form $\pi = \mathcal{N}(0, V)$. In such cases, we can compute the ground-truth optimal variational approximation either analytically (because the distribution belongs to the mean-field variational family and hence $q_* = \pi$) or numerically using deterministic optimization (since the KL divergence between Gaussians is available in closed form). Specifically, we consider the following covariances that aid in assessing our framework across a range of condition numbers from 1 to around 9000:

- Identity covariance: $V = I$
- Diagonal non-identity covariance: $V_{ij} = j\mathbb{1}[i = j]$
- Uniform covariance with correlation 0.8: $V_{ij} = \mathbb{1}[i = j] + 0.8\mathbb{1}[i \neq j]$
- Banded covariance with maximum correlation 0.8: $V_{ij} = \mathbb{1}[i = j] + 0.8^{|i-j|}\mathbb{1}[i \neq j]$
- Diagonal non-identity banded covariance with maximum correlation 0.8: $V_{ij} = j\mathbb{1}[i = j] + 0.8^{|i-j|}\mathbb{1}[i \neq j]$
- Diagonal identity (except first entry) uniform covariance with maximum correlation 0.8: $V_{ij} = 10001[i = j = 1] + \mathbb{1}[i = j \neq 1] + 0.8\mathbb{1}[i \neq j]$
- Diagonal identity (except first entry) banded covariance with maximum correlation 0.8: $V_{ij} = 10001[i = j = 1] + \mathbb{1}[i = j \neq 1] + 0.8^{|i-j|}\mathbb{1}[i \neq j]$

In our selection, we specifically included diagonal identity matrices (with the exception of the first entry) combined with either uniform or banded covariance structures, showcasing a maximum covariance of 0.8. This setting results in weaker correlation between the first component and the others. This choice was strategic to achieve higher condition numbers (around 5000 and 9000 respectively), given that correlations set at 0.8 or $0.8^{|i-j|}\mathbb{1}[i \neq j]$ yield condition numbers around 400 and 80, respectively.

Figures 7, C.8 and C.9 show the comparison of RABVI to FASO, Stan's ADVI implementation, SGD with exponential decay learning rate (SGD-ED), Adam with exponential decay and cosine learning rates (Adam-ED, Adam-C), RMSProp with exponential decay and cosine learning rates (RMSProp-ED, RMSProp-C), and fixed-learning rate versions of RMSProp, Adam, Windowed Adagrad (WAdagrad). The findings demonstrate that RABVI consistently outperforms ADVI, SGD-ED, both adaptive learning rate versions of RMSProp, and the fixed-learning rate methods in a majority of the cases. While Adam and SGD-ED occasionally reach performance levels similar to RABVI, they tend to converge more slowly and with less reliability. Additionally, despite Adam-ED and Adam-C closely matching RABVI's performance on most problems, they lack a dependable mechanism for determining when to terminate the optimization at a desired accuracy level. Although a validation data set can be used, this requires the availability of such a data set and would allow for control of the approximation accuracy. On the other hand, by varying the accuracy threshold ξ, the quality of the final RABVI approximation $\hat{q}_*$ also varies such that $\mathrm{SKL}(q_*, \hat{q}_*)^{1/2} \approx \xi$.

To demonstrate the flexibility of our framework, we used RABVI with a variety of optimization methods: RMSProp, avgRMSProp, avgAdam, natural gradient descent (NGD), and stochastic quasi-Newton (SQN). See Appendices B.5 and B.6 for details. Figure C.10 shows that avgAdam and avgRMSProp optimization methods have a similar improvement in symmetrized KL divergence between optimal and estimated variational approximation for all cases. NGD is not stable for the diagonal non-identity covariance structure and SQN does not perform well with uniform covariance structure. Even though RMSProp shows an improvement in accuracy for large step sizes, accuracy does not improve as the step size decreases.



(a) uncorrelated $d = 100$

(b) uncorrelated $d = 500$

(c) uniform correlated $d = 100$

(d) banded correlated $d = 100$

Figure 7: Accuracy comparison of variational inference algorithms using Gaussian targets, where accuracy is measured in terms of the square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the termination rule trigger points of FASO and RABVI. Iterate averages for Adam, RMSProp, and WAdagrad computed at every 200th iteration using a window size of 20% of iterations.

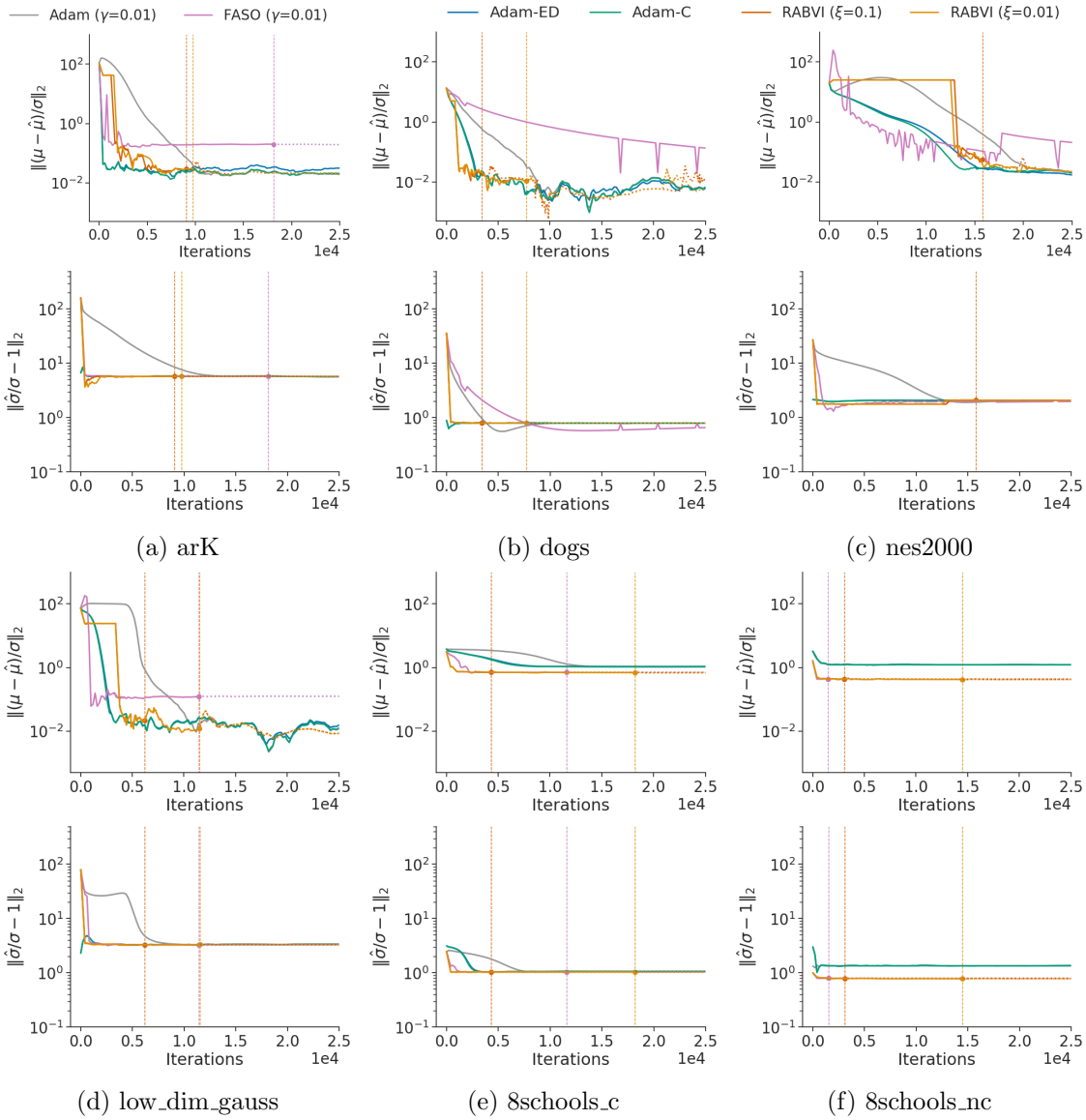
7.2 Reliability Across Applications

To validate the robustness and reliability of RABVI across realistic use cases, we consider 18 diverse data set/model pairs found in the `posteriordb` package[3] (see Appendix C for details). The `posteriordb` package contains a wide range of real-world data and models and is specifically designed to provide realistic performance evaluations of approximate posterior inference algorithms. The accuracy was computed based on ground-truth estimates obtained using the posterior draws included in `posteriordb` package if available. Otherwise, we ran Stan's dynamic HMC algorithm (Stan Development Team, 2020) to obtain the ground truth (4 chains for 50,000 iterations each). To stabilize the optimization, we initialize the variational parameter estimates using RMSProp for the initial learning rate only. A comparison across optimization methods validates our choice of avgAdam over alternatives (Fig. C.11).

Comparison to alternative optimization methods. To evaluate RABVI's effectiveness in real-world applications, we compared it against alternative optimization methods with both fixed and adaptive learning rate schedules. Based on the results described in Section 7.1, we opt to compare to Adam using either a fixed, exponential decay, or cosine learning rate schedule since they perform best overall in the Gaussian target experiments. Additionally, we include FASO, which used avgAdam, as another benchmark. Figure 8 shows RABVI is more consistent than all the alternative methods. While these methods sometimes matched RABVI's performance, RABVI's ability to identify an appropriate stopping point contributes to its overall efficiency, setting it apart from the competition.

Accuracy and robustness. First, we investigate the accuracy and algorithmic robustness of RABVI. In terms of robustness, Figures 9, C.12 and C.13 validate our termination criteria since after reaching the termination point there is no considerable improvement in the accuracy for most of the models and data sets. While in many cases the mean estimates are quite accurate, the standard deviation estimates tended to be poor, which is consistent with typical behavior of mean-field approximations. To examine whether RABVI can achieve more accurate results with more flexible variational families, we conduct the same experiment using multivariate Gaussian approximation family and normalizing flow family using real-NVP flow with 2 hidden layers, 8 hidden units, and 3 coupling layers (Dinh et al., 2017). We employ FASO in our real-NVP experiments because the complexity of the approximation family prevents us from obtaining a closed form for the SKL divergence, which is necessary for computing the termination rule in RABVI. In some cases the accuracy of the mean and/or standard deviations estimates improve (*bball_0*, *dogs_log*, *8schools_c*, *hudson_lynx*, *hmm_example*, *nes2000*, and *sblrc*). However, the results are inconsistent and sometimes worse due to the higher-dimensional, more challenging optimization problem. Our findings underscore the necessity of supplementing an improved optimization framework like RABVI with diagnostics for assessing the accuracy of the posterior approximation (Yao et al., 2018; Huggins et al., 2020; Wang et al., 2023c).

3. `https://github.com/stan-dev/posteriordb`



(a) arK (b) dogs (c) nes2000

(d) low_dim_gauss (e) 8schools_c (f) 8schools_nc

Figure 8: Accuracy comparison of variational inference algorithms using `posteriordb` models and data sets, where accuracy is measured in terms of relative mean error (top) and relative standard deviation error (bottom). The vertical lines indicate the termination rule trigger points of FASO and RABVI. The iterate average for Adam is computed at every 200^{th} iteration using a window size of 20% of iterations.

Comparison to MCMC. We additionally benchmarked the runtime and accuracy of RABVI to `Stan`'s dynamic HMC algorithm, for which we ran 1 chain for 25,000 iterations including 5,000 warmup iterations. We measure runtime in terms of the number of likelihood evaluations and compared the relative error between the methods at the RABVI termination rule trigger point or final likelihood evaluation of HMC (whichever comes first). Figures 10, C.15 and C.16 show that RABVI tends to provide similar or better posterior mean estimates (the exceptions are *gp_pois_regr*, *hudson_lynx*, and *sblrc*). However, the RABVI standard deviation estimates tend to be less accurate even when using the full-rank Gaussian variational family. This could be because the optimization of full-rank Gaussian is more challenging having more variational parameters to estimate (Bhatia et al., 2022). Figure C.17 shows that, in terms of the MMD, the HMC approximation is closer to the target than BBVI as one would expect. Overall, the MMD values for RABVI are reasonably small.

We also compared the 95% quantiles posterior under coverage error between RABVI and FASO methods using different approximation families and MCMC. Figure C.18 shows that HMC and real-NVP flows do not undercover the posterior. However, the mean-field and full-rank Gaussian families do.

7.3 Robustness to Tuning Parameters: Ablation Study

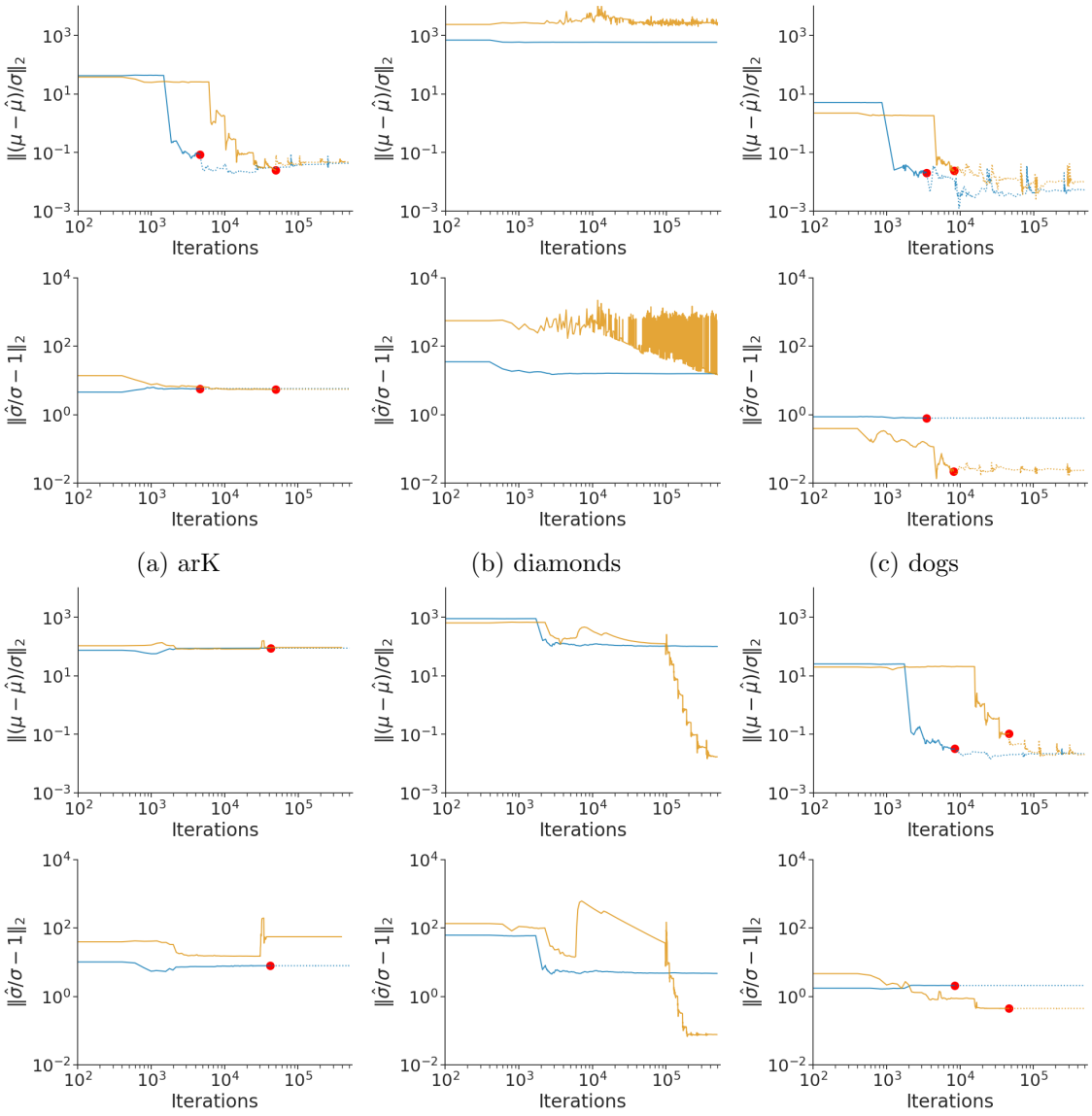
To validate the robustness of RABVI to different choices of algorithm tuning parameters, we consider the Gaussian targets and two *posteriordb data set/model pairs: dogs (logistic mixed-effects model) and arK (AR(5) time-series model)*. We vary one tuning parameter while keeping the recommended default values for all others. We consider the following values for each parameter (default in bold):

- initial learning rate γ_0: $0.01, 0.1, \mathbf{0.3}, 0.5$
- minimum window size $W_{\min}$: $100, \mathbf{200}, 300, 500$
- initial iterate average relative error threshold ε_0: $0.05, \mathbf{0.1}, 0.2, 0.5$
- inefficiency threshold τ: $0.1, 0.5, \mathbf{1.0}, 1.2$
- Monte Carlo samples M: $1, 5, \mathbf{10}, 15, 25$.

We repeat each experimental condition 10 times to verify the robustness of different initializations of the variational parameters. Figures 11 and C.19 to C.25 suggest that overall the accuracy and runtime of RABVI is not too sensitive to the choice of the tuning parameters. However, extreme tuning parameter choices (for example, $\gamma = 0.01$ or $M \leq 5$) can lead to longer runtimes.

8. Discussion

As we have shown through both theory and experiments, RABVI, our stochastic optimization framework for black-box variational inference, delivers a number of benefits compared to existing approaches:



(a) arK (b) diamonds (c) dogs

(d) gp_pois_regr (e) sblrc (f) nes2000

Figure 9: Accuracy of mean-field (blue) and full-rank (orange) Gaussian family approximations for selected `posteriordb` data/models, where accuracy is measured in terms of relative mean error (top) and relative standard deviation error (bottom). The red dots indicate where the termination rule triggers.

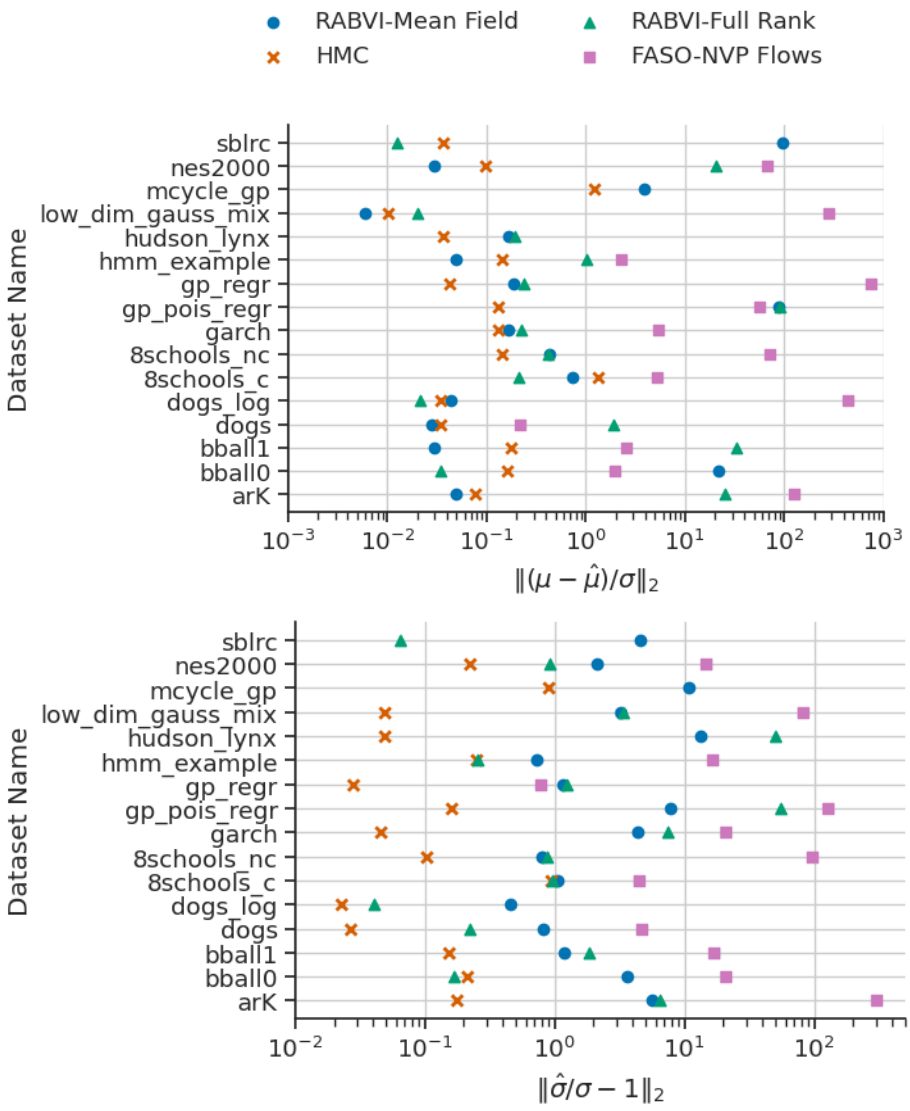
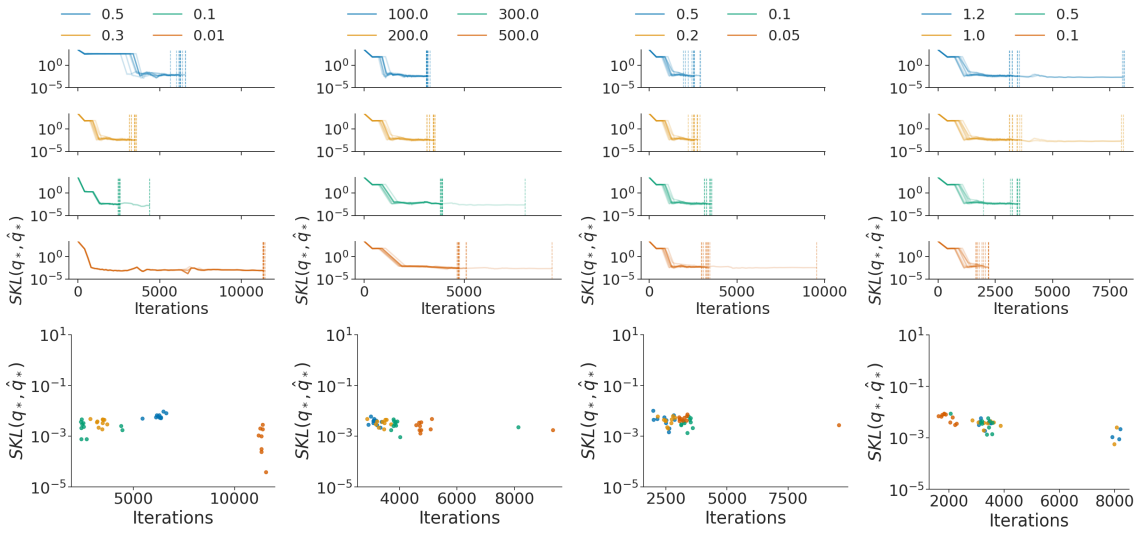


Figure 10: Results of RABVI with mean-field Gaussian and full-rank Gaussian family and FASO with real NVP flows comparison to dynamic HMC at the same computational cost (likelihood evaluations) in terms of relative mean error (top) and relative standard deviation error (bottom).



(a) γ_0 (b) W_{min} (c) ε_0 (d) τ

Figure 11: Robustness to tuning parameters (a) initial learning rate γ_0, (b) minimum window size $W_{\min}$, (c) initial iterate average relative error threshold ε_0, and (d) inefficiency threshold τ using *dogs* data set from `posteriordb` package. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The transparent lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.

- The user only needs to, at most, adjust a small number of tuning parameters which empowers the user to intuitively control and trade off computational cost and accuracy. Moreover, RABVI is robust, both in terms of accuracy and computational cost, to small changes in all tuning parameters.

- Our framework can easily incorporate different stochastic optimization methods such as adaptive versions, natural gradient descent, and stochastic quasi-Newton methods. In practice, we found that the averaged versions of RMSProp and Adam we propose perform particularly well. However the performance of RABVI will benefit from future innovations in stochastic optimization methodology.

- RABVI allows for any choice of tractable variational family and stochastic gradient estimator. For example, in many cases we find accuracy improves when using the full-rank Gaussian variational family rather than the mean-field one.

Our empirical results also highlight some of the limitations of BBVI, which sometimes is less accurate than dynamic HMC when given equal computational budgets. However, BBVI

can be further sped up using, for example, data subsampling when the data set size is large (which was not the case for the `posteriodb` data sets from our experiments).

Acknowledgments

M. Welandawe and J. H. Huggins were supported by the National Institute of General Medical Sciences of the National Institutes of Health under grant number R01GM144963 as part of the Joint NSF/NIGMS Mathematical Biology Program. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. A. Vehtari was supported by Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI, and M. R. Andersen was supported by Innovation Fund Denmark (grant number 8057-00036A).

Appendix A. Proofs

A.1 Proof of Proposition 6

Since the KL divergence of mean-field Gaussians factors across dimensions, without loss of generality we consider the case of dim = 1. The symmetrized KL divergence between two Gaussians is

$$\begin{aligned}\mathrm{SKL}(\lambda_1,\lambda_2) &= \frac{1}{2}\left\{e^{2\psi_1-2\psi_2}+e^{2\psi_2-2\psi_1}+(\tau_1-\tau_2)^2(e^{-2\psi_1}+e^{-2\psi_2})-2\right\}\\ &= \frac{1}{2}\left\{e^{2\psi_1-2\psi_2}+e^{2\psi_2-2\psi_1}+(\tau_1-\tau_2)^2e^{-2\psi_1}(1+e^{2\psi_1-2\psi_2})-2\right\}.\end{aligned}$$

Recall our assumption that for some constants $A=(A_\tau,A_\psi)$ and $B=(B_\tau,B_\psi)$,

$$\bar{\lambda}_\gamma = \lambda_* + A\gamma + B\gamma^2 + o(\gamma^2).$$

Hence

$$\begin{aligned}e^{2\bar{\psi}_\gamma-2\psi_*} &= 1+2A_\psi\gamma+2B_\psi\gamma^2+\frac{1}{2}(2A_\psi\gamma+2B_\psi\gamma^2)^2+o(\gamma^2)\\ &= 1+2A_\psi\gamma+2B_\psi\gamma^2+2A_\psi^2\gamma^2+o(\gamma^2)\end{aligned}$$

and similarly $e^{2\psi_*-2\bar{\psi}_\gamma}=1-2A_\psi\gamma-2B_\psi\gamma^2+2A_\psi^2\gamma^2+o(\gamma^2)$. Therefore $e^{2\bar{\psi}_\gamma-2\psi_*}+e^{2\psi_*-2\bar{\psi}_\gamma}-2=4A_\psi^2\gamma^2+o(\gamma^2)$ and $(\bar{\tau}_\gamma-\tau_*)^2=A_\tau^2\gamma^2+o(\gamma^2)$. Putting these results together, we have

$$\begin{aligned}\mathrm{SKL}(\bar{\lambda}_\gamma,\lambda_*) &= \frac{1}{2}\Big[4A_\psi^2\gamma^2+o(\gamma^2)+A_\tau^2\gamma^2e^{-\psi_*}\{2-2A_\psi\gamma-2B_\psi\gamma^2+2\gamma^2A_\psi^2+o(\gamma^2)\}\Big]\\ &= (2A_\psi^2+A_\tau^2e^{-2\psi_*})\gamma^2+o(\gamma^2).\end{aligned}$$

Similarly, $\mathrm{SKL}(\bar{\lambda}_\gamma,\bar{\lambda}_{\gamma'})=(\gamma-\gamma')^2(2A_\psi^2+A_\tau^2e^{-2\psi_*})+o(\gamma^2)$ as long as $\gamma'=O(\gamma)$. In other words, letting $\bar{\lambda}_0:=\lambda_*$, we have the relation

$$\mathrm{SKL}(\bar{\lambda}_\gamma,\bar{\lambda}_{\gamma'})=C(\gamma-\gamma')^2+o(\gamma^2)$$

for some unknown constant C.

A.2 Proof of Proposition 7

Define the Fisher information matrix $F_\lambda=\mathbb{E}_{\theta\sim q_\lambda}\left[-\nabla_\lambda^2\log\{q_\lambda(\theta)\}\right]$. Since $\lambda\mapsto q_\lambda(\theta)$ is three-times differentiable, locally the KL divergence behaves in a quadratic form (Amari, 2016):

$$\mathrm{SKL}(\lambda_1,\lambda_2)=\frac{1}{2}\left\{(\lambda_1-\lambda_2)^\top F_{\lambda_2}(\lambda_1-\lambda_2)+(\lambda_2-\lambda_1)^\top F_{\lambda_1}(\lambda_1-\lambda_2)\right\}+o(\|\lambda_1-\lambda_2\|^2).$$

Moreover, by taking first-order Taylor expansion, we have

$$F_{\lambda_i}=F_{\lambda_*}+O(\|\lambda_*-\lambda_i\|_2).$$

If $\kappa = 1$, recall that for some constant vectors A and B,

$$\bar{\lambda}_\gamma = \lambda_* + A\gamma + B\gamma^2 + o(\gamma^2).$$

Assume that $\gamma_1 = \gamma$ and $\gamma_2 = O(\gamma)$, so $\|\lambda_* - \bar{\lambda}_{\gamma_i}\|_2 = O(\gamma)$. For $\kappa' > 0$, let $\epsilon_{\kappa'} := \gamma_1^{\kappa'} - \gamma_2^{\kappa'}$. Then we have

$$\begin{aligned}
\mathrm{SKL}(\bar{\lambda}_{\gamma_1}, \bar{\lambda}_{\gamma_2}) = \frac{1}{2}\Bigg[&(A\epsilon_1 + B\epsilon_2 + o(\gamma^2))^\top (F_{\lambda_*} + O(\gamma))(A\epsilon_1 + B\epsilon_2 + o(\gamma^2)) \\
&+ (-A\epsilon_1 - B\epsilon_2 - o(\gamma^2))^\top (F_{\lambda_*} + O(\gamma))(-A\epsilon_1 - B\epsilon_2 - o(\gamma^2))\Bigg] \\
&+ o(\|A\epsilon_1 + B\epsilon_2 + o(\gamma^2)\|_2^2) \\
&= A^\top F_{\lambda_*} A\epsilon_1^2 + o(\gamma^2) \\
&= C(\gamma_1 - \gamma_2)^2 + o(\gamma^2),
\end{aligned}$$

where $C = A^\top F_{\lambda_*} A$.

If $\kappa \in [1/2, 1)$, recall that for some constant vectors Λ and A,

$$\bar{\lambda}_\gamma = \lambda_* + \Lambda\gamma^\kappa + A\gamma + o(\gamma^{2\kappa}).$$

Assume that $\gamma_1 = \gamma$ and $\gamma_2 = O(\gamma)$, so $\|\lambda_* - \bar{\lambda}_{\gamma_i}\|_2 = O(\gamma^\kappa)$. Then we have

$$\begin{aligned}
\mathrm{SKL}(\bar{\lambda}_{\gamma_1}, \bar{\lambda}_{\gamma_2}) = \frac{1}{2}\Bigg[&(\Lambda\epsilon_\kappa + A\epsilon_1 + o(\gamma^{2\kappa}))^\top (F_{\lambda_*} + O(\gamma^\kappa))(\Lambda\epsilon_\kappa + A\epsilon_1 + o(\gamma^{2\kappa})) \\
&+ (-\Lambda\epsilon_\kappa - A\epsilon_1 - o(\gamma^{2\kappa}))^\top (F_{\lambda_*} + O(\gamma^\kappa))(-\Lambda\epsilon_\kappa - A\epsilon_1 - o(\gamma^{2\kappa}))\Bigg] \\
&+ o(\|\Lambda\epsilon_\kappa + A\epsilon_1 + o(\gamma^{2\kappa})\|_2^2) \\
&= \Lambda^\top F_{\lambda_*} \Lambda\epsilon_\kappa^2 + 2\Lambda^\top F_{\lambda_*} A\epsilon_\kappa\epsilon_1 + A^\top F_{\lambda_*} A\epsilon_1^2 + o(\gamma^{2\kappa}) \\
&= \Lambda^\top F_{\lambda_*} \Lambda\epsilon_\kappa^2 + O(\gamma^{1+\kappa}) + o(\gamma^{2\kappa}) \\
&= C(\gamma_1^\kappa - \gamma_2^\kappa)^2 + o(\gamma^{2\kappa}),
\end{aligned}$$

where $C = \Lambda^\top F_{\lambda_*} \Lambda$.

A.3 Symmetric KL Divergence Termination Rule

We can use Proposition 7 to derive a termination rule. If γ is the current learning rate, the previous learning rate was γ/ρ. Ignoring $o(\gamma^{2\kappa})$ terms, we have

$$\delta_\gamma := \mathrm{SKL}(\bar{\lambda}_{\gamma/\rho}, \bar{\lambda}_\gamma) = C\gamma^{2\kappa}(1/\rho^\kappa - 1)^2.$$

Therefore, we can estimate p and C using a regression model of the form

$$\log \delta_\gamma = \log C + 2\log(1/\rho^\kappa - 1) + 2p \log \gamma.$$

Given estimates $\hat{\kappa}$ and $\hat{C}$, we can estimate $\mathrm{SKL}(\bar{\lambda}_\gamma, \lambda_*) \approx \hat{C}\gamma^{2\hat{\kappa}}$.

A.4 Proof of Proposition 8

Since for $|x| < 1/2$ it holds that $|\exp(x) - 1| \leq 1.5|x|$, we have

$$\frac{|\hat{\sigma}_i - \bar{\sigma}_i|}{\bar{\sigma}_i} = |\exp(\hat{\psi}_i - \bar{\psi}_i) - 1| \leq 1.5\varepsilon$$

and

$$\frac{|\hat{\tau}_i - \bar{\tau}_i|}{\bar{\sigma}_i} \leq \varepsilon' \hat{\sigma}_i / \bar{\sigma}_i \leq \varepsilon'(1 + 1.5\varepsilon) \leq 1.75\varepsilon.$$

A.5 Proof of Proposition 9

In the optimal case, the total cost for the iterations used for iterate averaging is

$$\begin{aligned}\mathrm{OPT} &= C_O W_{\mathrm{opt}} + C_E W_{\mathrm{conv}} + C_E W_{\mathrm{opt}} \\ &= C_E(r W_{\mathrm{opt}} + W_{\mathrm{conv}} + W_{\mathrm{opt}}).\end{aligned}$$

On the other hand, if checking at window sizes $W_j := \chi^j W_{\mathrm{conv}}$ $(j = 1, 2, \dots)$, then the window size at which convergence will be detected is $j_* := \lceil \log(W_{\mathrm{opt}}/W_{\mathrm{conv}})/\log(\chi) \rceil$. In particular,

$$W_{j_*} \leq \chi W_{\mathrm{opt}}.$$

Therefore we can bound the actual computational cost as

$$\begin{aligned}&C_O W_{j_*} + C_E W_{\mathrm{conv}} \log W_{\mathrm{conv}} + C_E \sum_{j=1}^{j_*} W_j \\ &\leq C_O \chi W_{\mathrm{opt}} + C_E W_{\mathrm{conv}} \log W_{\mathrm{conv}} + C_E \sum_{j=1}^{j_*} \chi^j W_{\mathrm{conv}} \\ &\leq C_O \chi W_{\mathrm{opt}} + C_E W_{\mathrm{conv}} \log W_{\mathrm{conv}} + C_E \frac{\chi(\chi^{j_*} - 1) W_{\mathrm{conv}}}{\chi - 1} \\ &\leq C_O \chi W_{\mathrm{opt}} + C_E W_{\mathrm{conv}} \log W_{\mathrm{conv}} + C_E \frac{\chi^2 W_{\mathrm{opt}}}{\chi - 1} \\ &= C_E \left[\chi r W_{\mathrm{opt}} + W_{\mathrm{conv}} + \frac{\chi^2}{\chi - 1} W_{\mathrm{opt}} \right].\end{aligned}$$

If $\chi = 2$, then the actual computational cost is bounded by

$$\begin{aligned}C_E \left[2r W_{\mathrm{opt}} + W_{\mathrm{conv}} + 4 W_{\mathrm{opt}} \right] &\leq 4 C_E \left[r W_{\mathrm{opt}} + W_{\mathrm{conv}} + W_{\mathrm{opt}} \right] \\ &= 4\,\mathrm{OPT}.\end{aligned}$$

On the other hand, we can minimize the total cost by solving $\chi_* := \arg\min_{\chi > 1} \{ r\chi + \chi^2/(\chi - 1) \} = \chi(r)$. Plugging this back in, we get the bound

$$\begin{aligned}&C_E \frac{2 + r + 2\sqrt{1 + r}}{1 + r} \left(r W_{\mathrm{opt}} + W_{\mathrm{conv}} \log W_{\mathrm{conv}} + W_{\mathrm{opt}} \right) \\ &= \frac{2 + r + 2\sqrt{1 + r}}{1 + r} \mathrm{OPT} \\ &< 4\,\mathrm{OPT}.\end{aligned}$$

Appendix B. Further Details

B.1 Effective Sample Size (ESS) and Monte Carlo Standard Error (MCSE)

Let $v^{(1)}, v^{(2)}, \ldots$ denote a stationary Markov chain, let $\bar{v} := \mathbb{E}[v^{(1)}]$ denote the mean at stationarity, and let $\hat{v} := K^{-1} \sum_{k=1}^{K} v^{(k)}$ denote the Monte Carlo estimate for $\bar{v}$. The (ideal) effective sample size is defined as

$$\mathrm{ESS}(K) := K/(1 + \sum_{k=1}^{\infty} \rho_k),$$

where ρ_k is the autocorrelation of the Markov chain at lag k. The ESS can be efficiently estimated using a variety of methods (Geyer, 1992; Vehtari et al., 2021). We write $\widehat{\mathrm{ESS}}(v^{(1:K)})$ to denote an estimator for $\mathrm{ESS}(K)$ based on the sequence $v^{(1:K)} := (v^{(1)}, \ldots, v^{(K)})$. The Monte Carlo standard error of $\hat{v}$ is given by

$$MCSE(\hat{v}) := \sigma(v^{(1)})/\mathrm{ESS}(K),$$

where $\sigma(v^{(1)})$ denote the standard deviation of the random variable $v^{(1)}$. Given the empirical standard deviation of $v^{(1)}, \ldots, v^{(K)}$, which we denote $\hat{\sigma}(v^{(1:K)})$, the MCSE can be approximated by

$$\widehat{\mathrm{MCSE}}(v^{(1:K)}) := \hat{\sigma}(v^{(1:K)})/\widehat{\mathrm{ESS}}(v^{(1:K)}).$$

B.2 Total Variation Distance and KL Divergence

For distributions η and ζ, $|I_{\eta,i,a,b} - I_{\zeta,i,a,b}| \leq \sqrt{\mathrm{KL}(\eta \mid \zeta)/2}$ for all $a < b$ and i. This guarantee follows from Pinsker's inequality, which relates the KL divergence to the total variation distance

$$d_{\mathrm{TV}}(\eta, \zeta) := \sup_{f:\|f\|_\infty \leq 1} \left| \int f(\theta)\eta(\mathrm{d}\theta) - \int f(\theta)\zeta(\mathrm{d}\theta) \right|,$$

where $\|f\|_\infty := \sup_\theta f(\theta) - \inf_\theta f(\theta)$. Specifically, Pinsker's inequality states that $d_{\mathrm{TV}}(\eta, \zeta) \leq \sqrt{\mathrm{KL}(\eta \mid \zeta)/2}$. Thus, small KL divergence implies small total variance distance, which implies the difference between expectations for any function f such that $\|f\|_\infty$ is small. In the case of interval probabilities, since $\|\mathbb{1}(\cdot \in [a, b])\|_\infty = 1$, it follows that

$$\begin{aligned} |I_{\eta,i,a,b} - I_{\zeta,i,a,b}| &= \left| \int \mathbb{1}(\theta \in [a, b])\eta(\mathrm{d}\theta) - \int \mathbb{1}(\theta \in [a, b])\zeta(\mathrm{d}\theta) \right| \\ &\leq \|\mathbb{1}(\cdot \in [a, b])\|_\infty \, d_{\mathrm{TV}}(\eta, \zeta) \\ &\leq \sqrt{\mathrm{KL}(\eta \mid \zeta)/2}. \end{aligned}$$

B.3 Adaptive SASA+

The SASA+ algorithm of Zhang et al. (2020) generalizes the approach of Yaida (2019). The main idea is to find an appropriate *invariant function* $\Delta(d, \lambda)$ that satisfies $\int \Delta(d, \lambda)\mu_\gamma(\mathrm{d}d, \mathrm{d}\lambda) = 0$. Yaida (2019) derived valid forms of Δ for specific choices of the descent direction,

while Zhang et al. (2020) showed that for any optimizer of the form Eq. (1) that is time-homogenous with $\gamma^{(k)} = \gamma$, the map $(d, \lambda) \mapsto 2\langle d, \lambda \rangle - \gamma \|d\|^2$ is a valid invariant function. The SASA+ algorithm proposed by Zhang et al. (2020) uses a hypothesis test to determine when the iterates are sufficiently close to stationarity. Let $\Delta^{(k)} := 2\langle d^{(k)}, \lambda^{(k)} \rangle - \gamma \left\| d^{(k)} \right\|^2$ and let $W = \lceil \varrho k \rceil$ denote the window size to use for checking stationarity. Once W is at least equal to a minimum window size $W_{\min}$, SASA+ uses $\Delta^{(k-W+1)}, \ldots, \Delta^{(k)}$ to carry out a hypothesis test, where the null hypothesis is that $\mathbb{E}[\Delta^{(k)}] = 0$.

We make several adjustments to reduce the number of tuning parameters and to make the remaining ones more intuitive. Note that the SASA+ convergence criterion requires the choice of three parameters: ϱ, $W_{\min}$, and the size of the hypothesis test α. Zhang et al. (2020) showed empirically and our numerical experiments confirmed that the choice of α has little effect and therefore does not need to be adjusted by the user. The choices for ϱ and $W_{\min}$, however, have a substantial effect on efficiency. If ϱ is too big, then early iterations that are not at stationarity will be included, preventing the detection of convergence. On the other hand, if ϱ is too small, then the total number of iterations must be large (specifically, greater than $W_{\min}/\varrho$) before the window size is large enough to trigger the first check for stationarity. Moreover, the correct choice of $W_{\min}$ will vary depending on the problem. If the iterates have large autocorrelation then $W_{\min}$ should be large, while if the autocorrelation is small or negative, then $W_{\min}$ can be small.

Our approach to determining the optimal window size instead relies on the effective sample size (ESS). W_{opt} that maximizes $\widehat{\text{ESS}}(W) := \widehat{\text{ESS}}(\Delta^{(k-W+1)}, \ldots, \Delta^{(k)})$, where k is the current iteration. To ensure reliability, we impose additional conditions on W_{opt}. First, we require that $\widehat{\text{ESS}}(W_{\text{opt}}) \geq N_{\min}$, a user-specified minimum effective sample size. Unlike $W_{\min}$, $N_{\min}$ has an intuitive and direct interpretation. The second condition is, when finding W_{opt}, the search over values of W is constrained to the lower bound of $N_{\min}$ (to ensure the estimator is sufficiently reliable) and the upper bound of $0.95k$ (to always allow for some "burn-in"). In practice we do not check all $W \in \{N_{\min}, \ldots, 0.95k\}$, but rather perform a grid search over 5 equally spaced values ranging from $N_{\min}$ to $0.95k$.

A slightly different version which may improve power is to instead define the multivariate invariant function $\vec{\Delta}(d, \lambda) = 2d \odot \lambda - d \odot d$, where we recall that $\odot$ denotes component-wise multiplication. In this case a multivariate hypothesis test such as Hotelling's T^2 test or the multivariate sign test, where a single effective sample size (for example, the median component-wise ESS) could be used. Alternatively, a separate hypothesis test for each of the m components could be used, with stationarity declared once all tests confirm stationarity (for example using a test size of α/m). While this approach might be more computationally efficient when m is large, it could come at the cost of test power.

B.4 Distance Based Convergence Detection

Pesme et al. (2020) proposed a distance-based diagnostic algorithm to detect the stationarity of the SGD optimization algorithm. The main idea behind this algorithm is to find the distance between the current iterate λ_k and the optimal variational parameter λ_*, $\|\eta_k\| := \|\lambda_k - \lambda_*\|$. Since the optimal variational parameter, λ_* is unknown, this distance cannot be directly observed. Therefore, they suggested using the distance between the current iterate

λ_k and the initial iterate of the current learning rate, $\|\Omega_k\| := \|\lambda_k - \lambda_0\|$ and showed that $\|\eta_k\|$ and $\|\Omega_k\|$ have a similar behavior.

Under the setting of the quadratic objective function with additive noise, they computed the behavior of the expectation of $\|\Omega_k\|^2$ in closed-form to detect the convergence of $\|\lambda_k - \lambda_0\|^2$. With the result, they have shown the asymptotic behavior of the $\mathbb{E}[\|\Omega_k\|^2]$ in the transient and stationary phases, where $\mathbb{E}[\|\Omega_k\|^2]$ has a slope greater than 1 and slope of 0 in a log-log plot respectively. Hence, the slope $S := \frac{\log\|\lambda_k-\lambda_0\|^2-\log\|\lambda_k/q-\lambda_0\|^2}{\log k-\log k/q}$ computed between iterations q^n and q^{n+1} for $q > 1$ and $n \geq n_0$ where $n_0 \in \mathcal{N}^*$ and if $S <$ thresh (thresh $\in (0, 2]$) then the convergence will be detected.

B.5 Stochastic Quasi-Newton Optimization

The algorithm of Liu and Owen (2021) provides a randomized approach of the classical quasi-Newton (QN) method that is known as the stochastic quasi-Newton (SQN) method. Even though, Liu and Owen (2021) uses randomized quasi-Monte Carlo (RMCQ) samples we use Monte Carlo (MC) samples to compute the gradient. In classical quasi-Newton optimization, the Newton update is

$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} - (\nabla^2_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}}))^{-1} \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}}),$$

where $\nabla^2_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}})$ is the Hessian matrix of $D_\pi(q_{\lambda^{(k)}})$. However, the computation cost of the Hessian matrix and its inverse is high, and it also requires a large amount of space. Therefore, we can use BFGS (discovered by Broyden, Fletcher, Goldfarb, and Shanno) method where it approximate the inverse of $\nabla^2_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}})$ using H_k at the kth iteration by initializing it with an identity matrix. Then the update is modified by

$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \gamma^{(k)} H^{(k)} \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}}),$$

where

$$H^{(k+1)} \leftarrow \Big(I - \frac{s_k y_k^\top}{s_k^\top y_k}\Big) H^{(k)} \Big(I - \frac{y_k s_k^\top}{s_k^\top y_k}\Big) + \frac{s_k s_k^\top}{s_k^\top y_k},$$

$s_k = \lambda^{(k+1)} - \lambda^{(k)}$, and $y_k = \nabla_{\lambda^{(k+1)}} D_\pi(q_{\lambda^{(k+1)}}) - \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}})$. Even using the above Hessian approximation $H^{(k)}$ will require a large space for storage. To overcome that problem we can use Limited-memory BFGS (L-BFGS) (Nocedal and Wright, 2006) that computes $H^{(k)} \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}})$ using m most recent (s_k, y_k) correction pairs. Liu and Owen (2021) use the stochastic quasi-Newton approach proposed by Chen et al. (2019) and they compute the correction pairs after every B iterations by computing the iterate average of parameters using the most recent B iterations. To compute the y_k, the gradients of the objective function are estimated using Monte Carlo samples that are independent of the samples used to compute the gradient in the update step.

B.6 Natural Gradient Descent Optimization

Khan and Lin (2017) proposed a natural gradient descent (NGD) approach for the variational inference that uses the information geometry of the variational distribution. Given

the variational distribution is an exponential family distribution

$$q_\lambda = h \exp(\phi^\top \lambda - A(\lambda))$$

the NGD update is

$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \gamma^{(k)} (F(\lambda^{(k)}))^{-1} \nabla_{\lambda^{(k)}} D_\pi(q_{\lambda^{(k)}}),$$

where $F(\lambda^{(k)})$ denotes the Fisher information matrix (FIM) of the distribution and λ denotes its natural parameter. Without directly computing the inverse of FIM Khan and Lin (2017) simplified the above update by using the relationship between natural parameter λ and expectation parameter $m = \mathrm{E}_q[\phi]$ of the exponential family

$$F(\lambda)^{-1} \nabla_\lambda D_\pi(q_\lambda) = \nabla_m D_\pi(q_m).$$

Therefore, the natural-gradient update can be simplified as

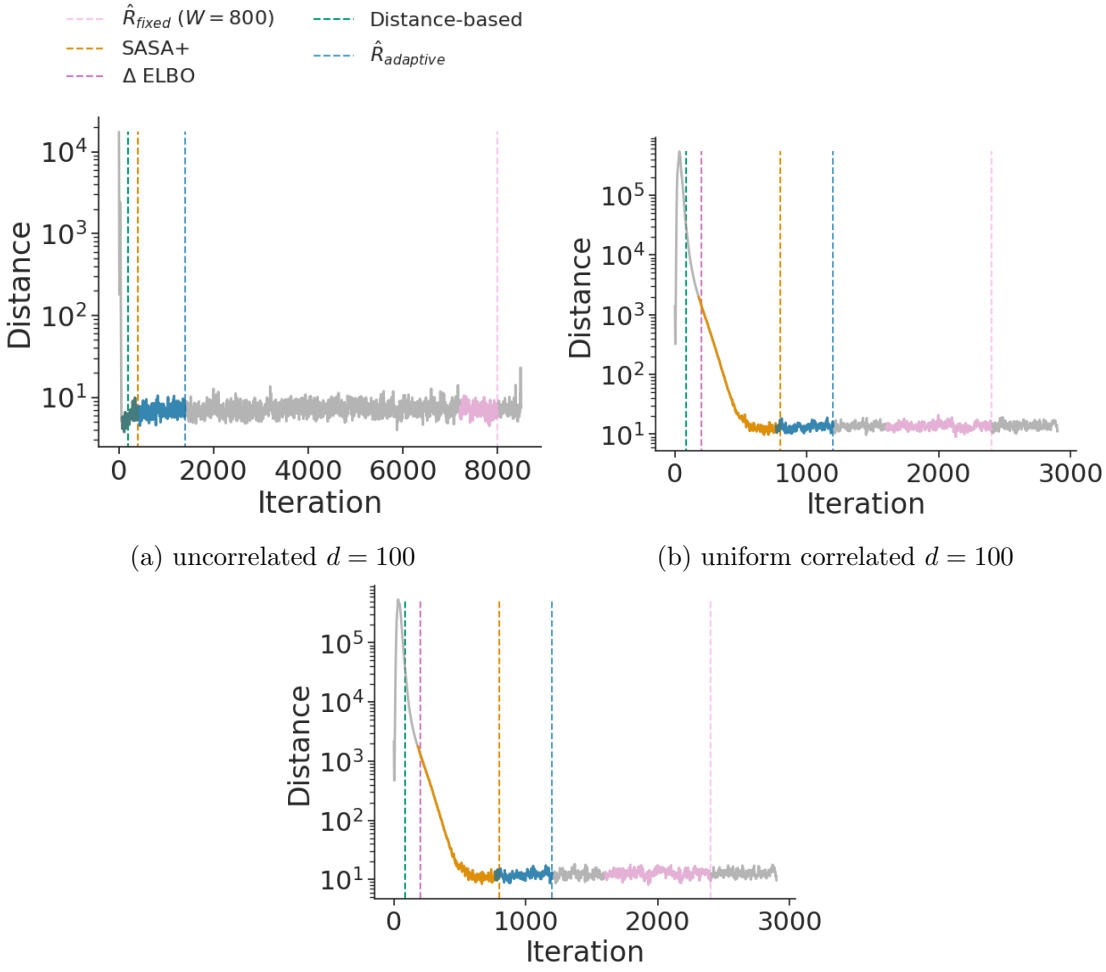
$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \gamma^{(k)} \nabla_{m^{(k)}} D_\pi(q_{m^{(k)}}).$$

This can be applied for the mean-field Gaussian variational family $q = \prod_{i=1}^{d} \mathcal{N}(\mu_i, \sigma_i^2)$ where we can define the natural parameters $\lambda_i = (\mu_i/\sigma_i^2, -0.5/\sigma_i^2)$ and expectation parameters $m = (\mu_i, \mu_i^2 + \sigma_i^2)$.

Appendix C. Additional Experimental Details and Results

Dataset	Short Name	D	Model Description
arK-arK	arK	7	AR(5) time series
bball_drive_event_0-hmm_drive_0	bball_0	8	Hidden Markov model
bball_drive_event_1-hmm_drive_1	bball_1	8	Hidden Markov model
diamonds-diamonds	diamonds	26	Log-Log
dogs-dogs	dogs	3	Logistic mixed-effects
dogs-dogs_log	dogs_log	2	Logarithmic mixed-effects
earnings-logearn_interaction	earnings	5	Log-linear
eight_schools-eight_schools_noncentered	8schools_nc	10	Non-centered hierarchical
eight_schools-eight_schools_centered	8schools_c	10	Centered hierarchical
garch-garch11	garch	4	GARCH(1,1) time series
gp_pois_regr-gp_pois_regr	gp_pois_regr	13	Gaussian process Poisson regression
gp_pois_regr-gp_regr	gp_regr	3	Gaussian process regression
hmm_example-hmm_example	hmm_example	6	Hidden Markov model
hudson_lynx-hare_lotka_volterra	hudson_lynx	8	Lotka-Volterra error
low_dim_gauss_mix-low_dim_gauss_mix	low_dim_gauss_mix	5	Two-dimensional Gaussian mixture
mcycle_gp-accel_gp	mcycle_gp	66	Gaussian process
nes2000-nes	nes2000	10	Multiple predictor linear
sblrc-blr	sblrc	6	Linear

Table C.1: Datasets of PosteriorDB Package

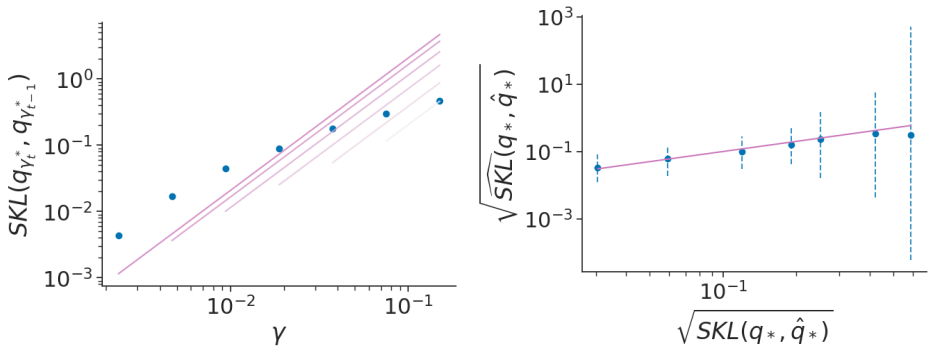


(a) uncorrelated $d = 100$

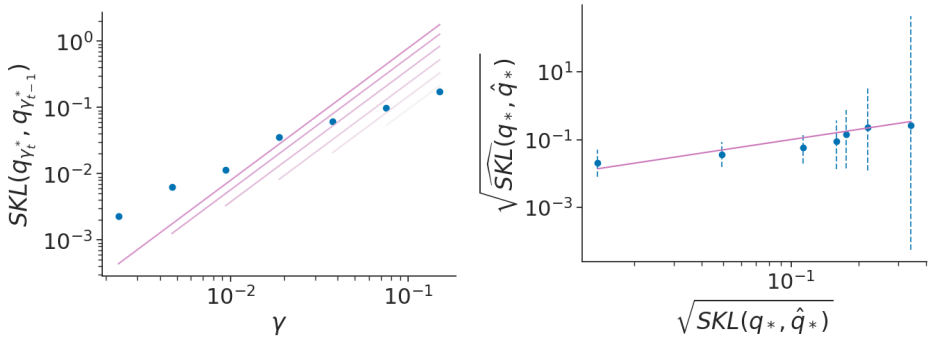
(b) uniform correlated $d = 100$

(c) banded correlated $d = 100$

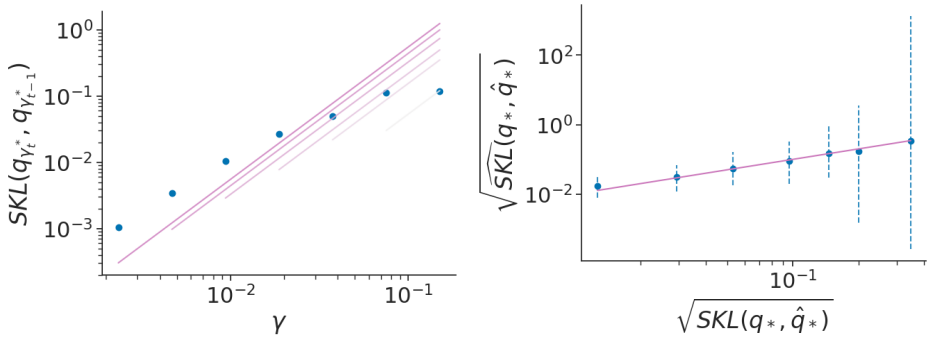
Figure C.1: Iteration number versus distance between iterate average and current iterate for Gaussian targets. The vertical lines indicate convergence detection trigger points and (for SASA+ and $\widehat{R}$) the colored portion of the accuracy values indicate they are part of the window used for convergence detection.



(a) uncorrelated $d = 500$

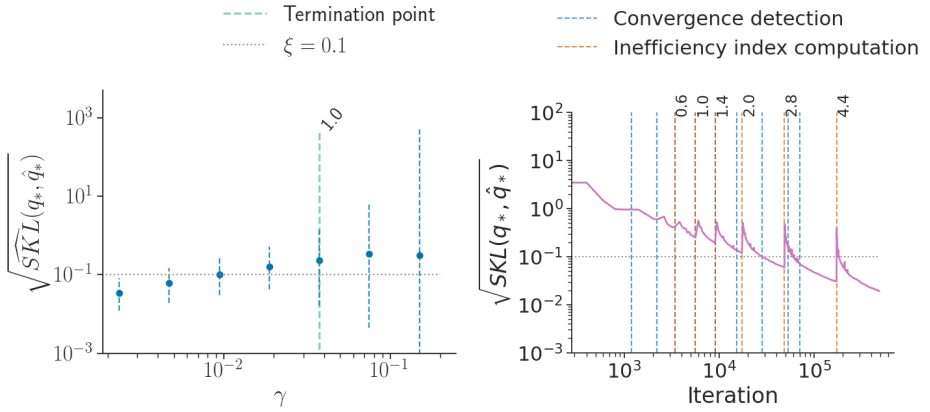


(b) uniform correlated $d = 100$

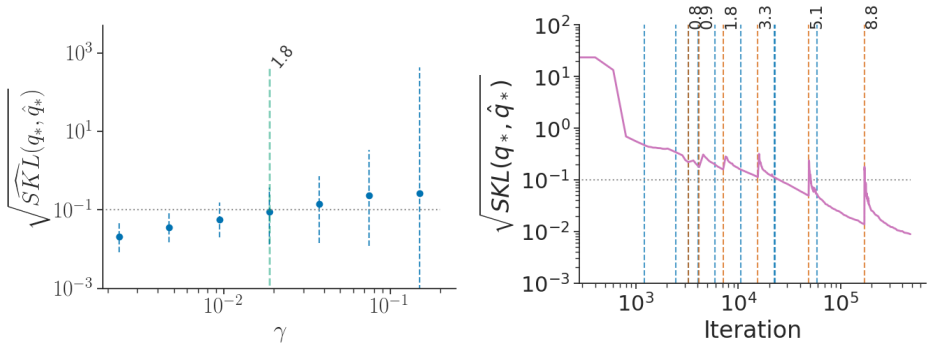


(c) banded correlated $d = 100$

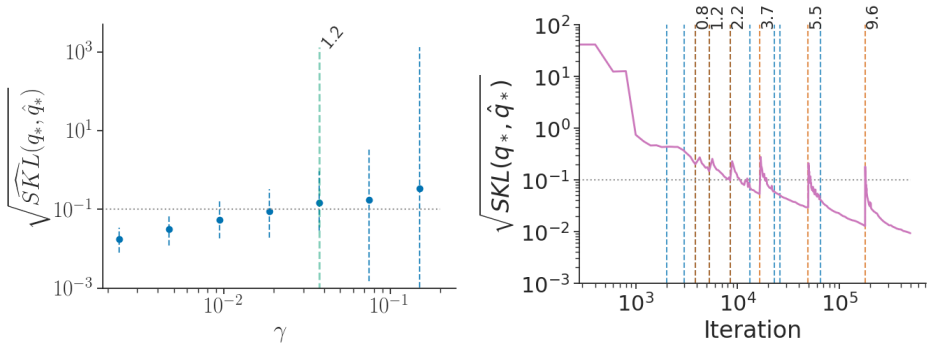
Figure C.2: Results for estimating the symmetrized KL divergence with avgAdam. **(left)** Learning rate versus symmetrized KL divergence of adjacent iterate averaged estimates of optimal variational distribution. The lines indicate the linear regression fits, with setting $\kappa = 1$. **(right)** Square root of true symmetrized KL divergence versus the estimated value with 95% credible interval. The uncertainty of the estimates decreases and remains well-calibrated as the learning rate decreases.



(a) uncorrelated $d = 500$

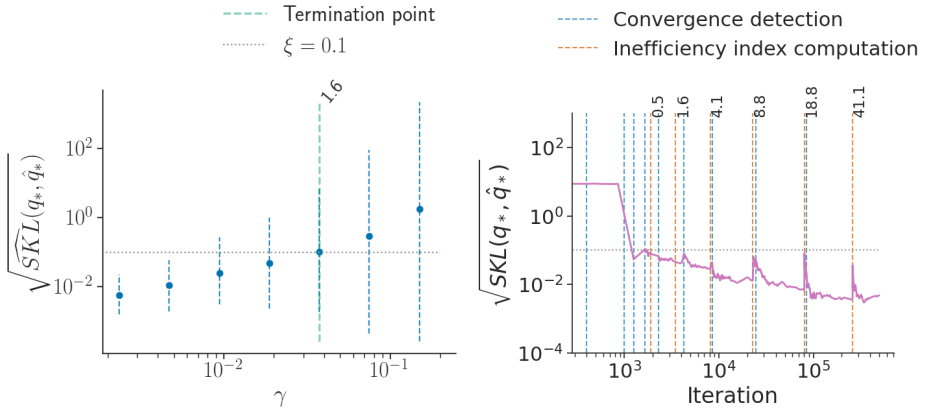


(b) uniform correlated $d = 100$

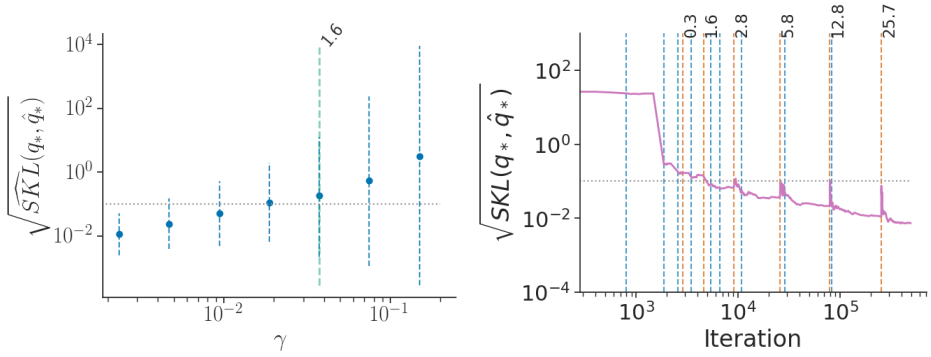


(c) banded correlated $d = 100$

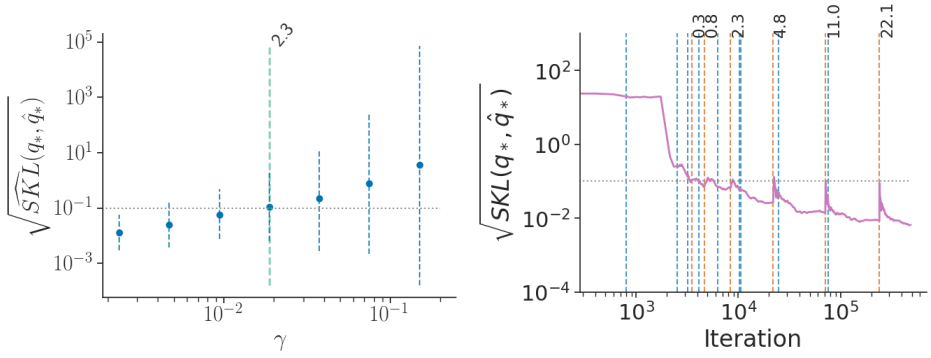
Figure C.3: Results of termination rule trigger point of Gaussian targets. **(left)** Learning rate versus square root of estimated symmetrized KL divergence with 95% credible interval (dashed blue line). The green vertical line indicates the termination rule trigger point with corresponding $\hat{\mathcal{I}}$ value. **(right)** Iterations versus square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the convergence detection points using $\hat{R}$ (blue) and inefficiency index computation ($\hat{\mathcal{I}}$) points (orange) with corresponding values.



(a) dogs

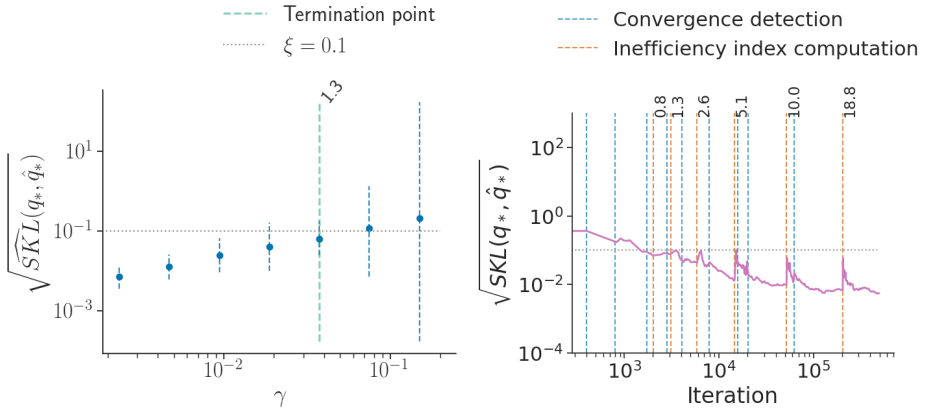


(b) arK

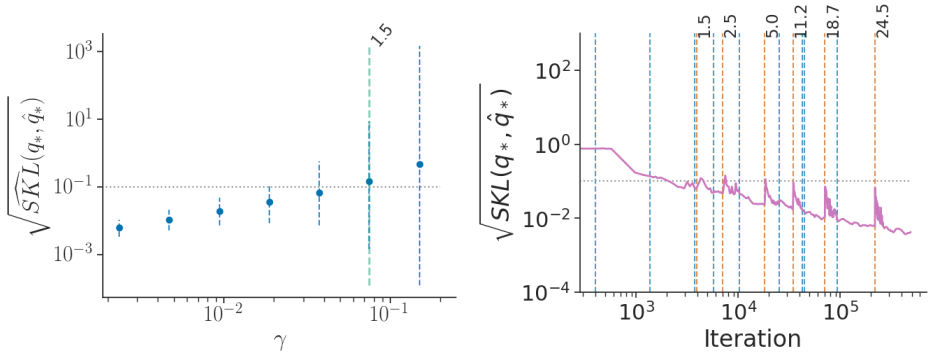


(c) nes2000

Figure C.4: Results of termination rule trigger point of `posteriordb` package datasets/models. **(left)** Learning rate versus square root of estimated symmetrized KL divergence with 95% credible interval (dashed blue line). The green vertical line indicates the termination rule trigger point with corresponding $\hat{\mathcal{I}}$ value. **(right)** Iterations versus square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the convergence detection points using $\hat{R}$ (blue) and inefficiency index computation ($\hat{\mathcal{I}}$) points (orange) with corresponding values.

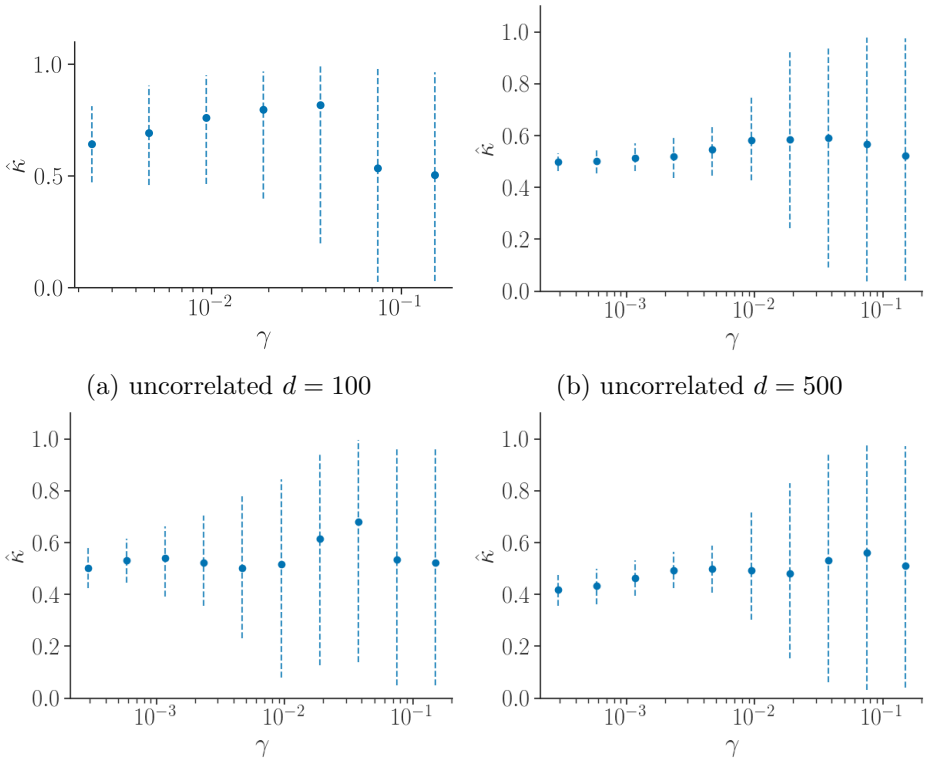


(a) 8schools_nc



(b) 8schools_c

Figure C.5: Results of termination rule trigger point of `posteriordb` package datasets/models. **(left)** Learning rate versus square root of estimated symmetrized KL divergence with 95% credible interval (dashed blue line). The green vertical line indicates the termination rule trigger point with corresponding $\hat{\mathcal{I}}$ value. **(right)** Iterations versus square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the convergence detection points using $\hat{R}$ (blue) and inefficiency index computation ($\hat{\mathcal{I}}$) points (orange) with corresponding values.



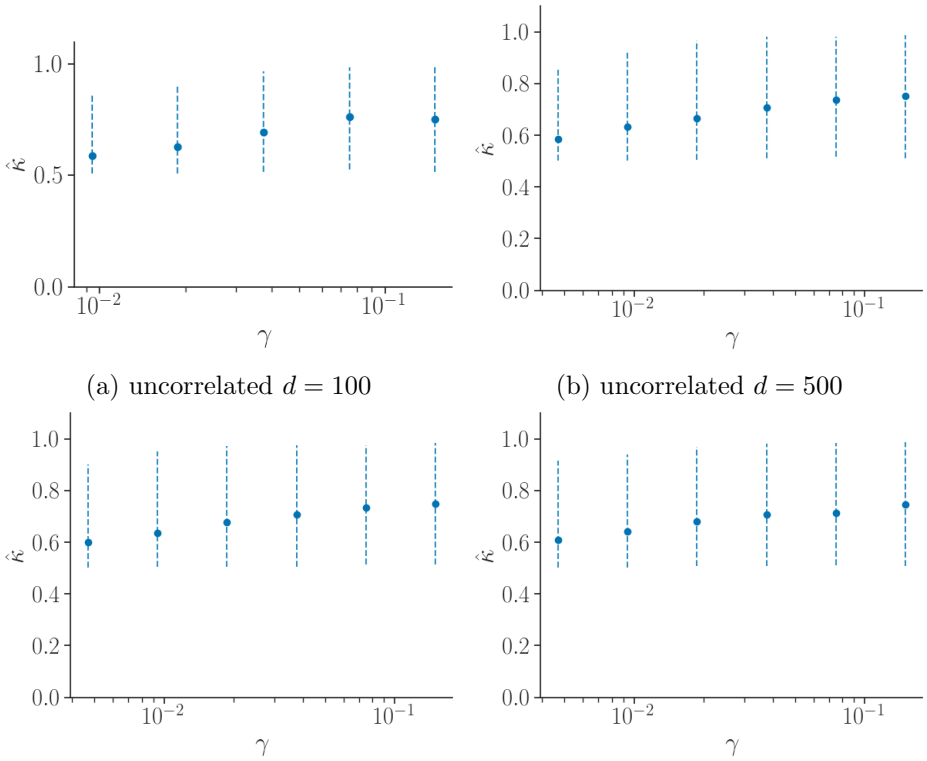
(a) uncorrelated $d = 100$

(b) uncorrelated $d = 500$

(c) uniform correlated $d = 100$

(d) banded correlated $d = 100$

Figure C.6: Learning rate versus $\hat{\kappa}$ for Gaussian targets using RMSProp with 95% credible interval. The estimates suggestion κ is approximately 0.5.



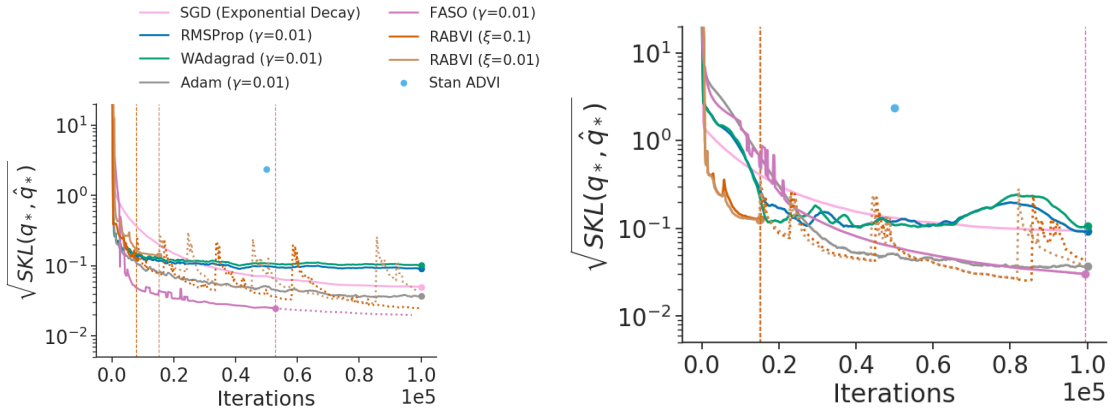
(a) uncorrelated $d = 100$

(b) uncorrelated $d = 500$

(c) uniform correlated $d = 100$

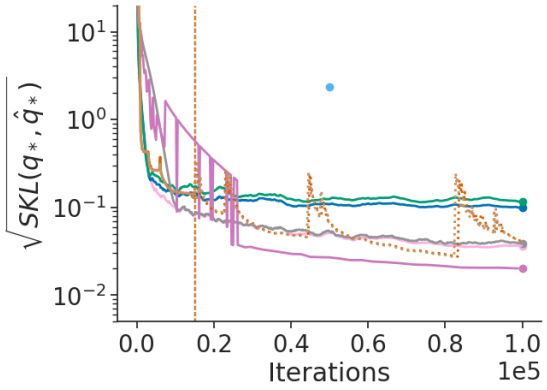
(d) banded correlated $d = 100$

Figure C.7: Learning rate versus $\hat{\kappa}$ for Gaussian targets using Adam with 95% credible interval. The estimates suggest κ is less than 0.8, with all point estimates close to 0.6.



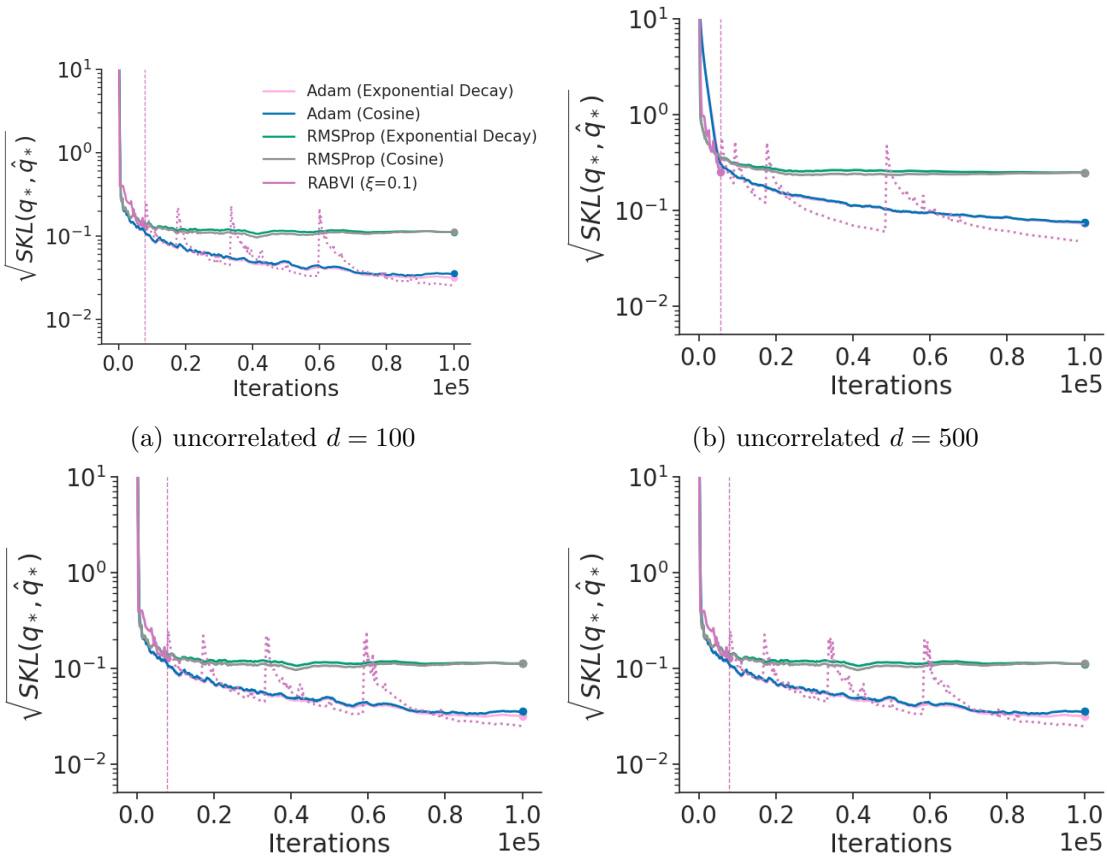
(a) diagonal non-identity banded correlated $d = 100$

(b) diagonal identity (except first entry) uniform correlated $d = 100$



(c) diagonal identity (except first entry) banded correlated $d = 100$

Figure C.8: Accuracy comparison of variational inference algorithms using Gaussian targets, where accuracy is measured in terms of the square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the termination rule trigger points of FASO and RABVI. Iterate averages for Adam, RMSProp, and WAdagrad computed at every 200th iteration using a window size of 20% of iterations.



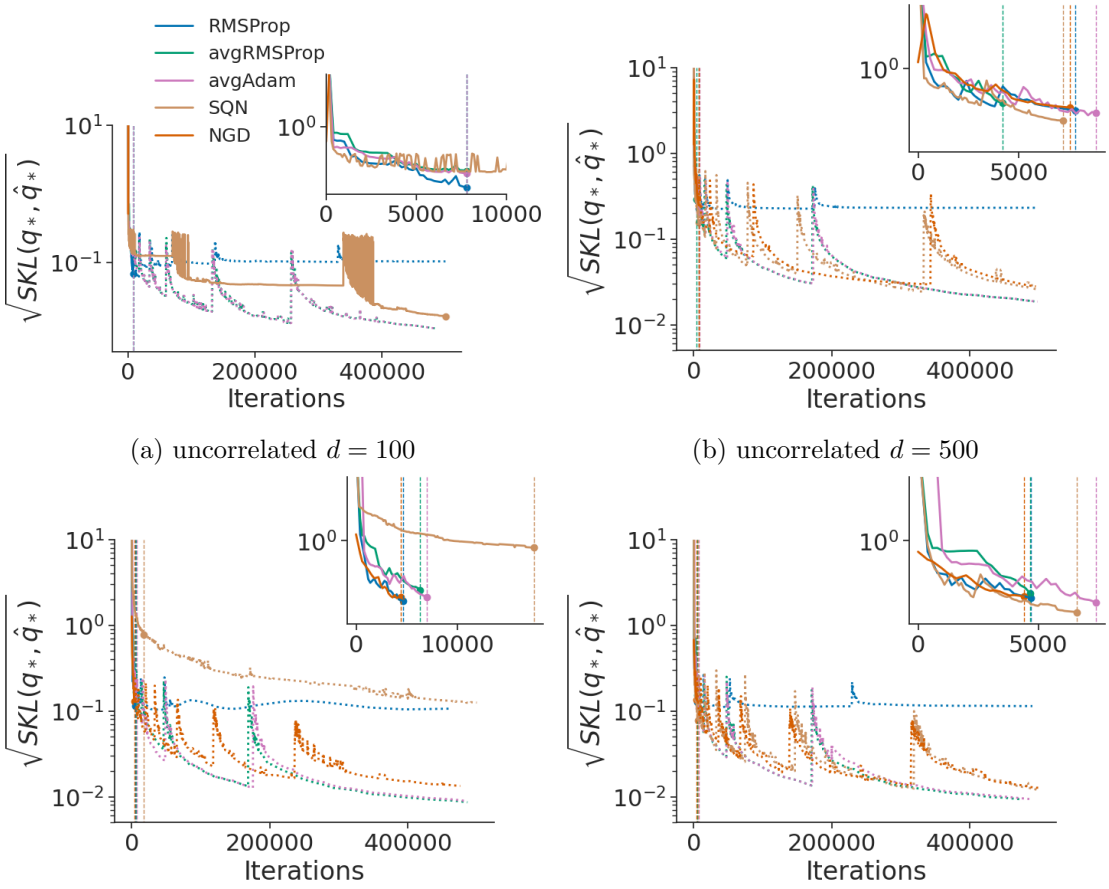
(a) uncorrelated $d = 100$

(b) uncorrelated $d = 500$

(c) uniform correlated $d = 100$

(d) banded correlated $d = 100$

Figure C.9: Accuracy comparison across learning rate schedules with Gaussian targets where accuracy is measured in terms of the square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the termination rule trigger points of RABVI.



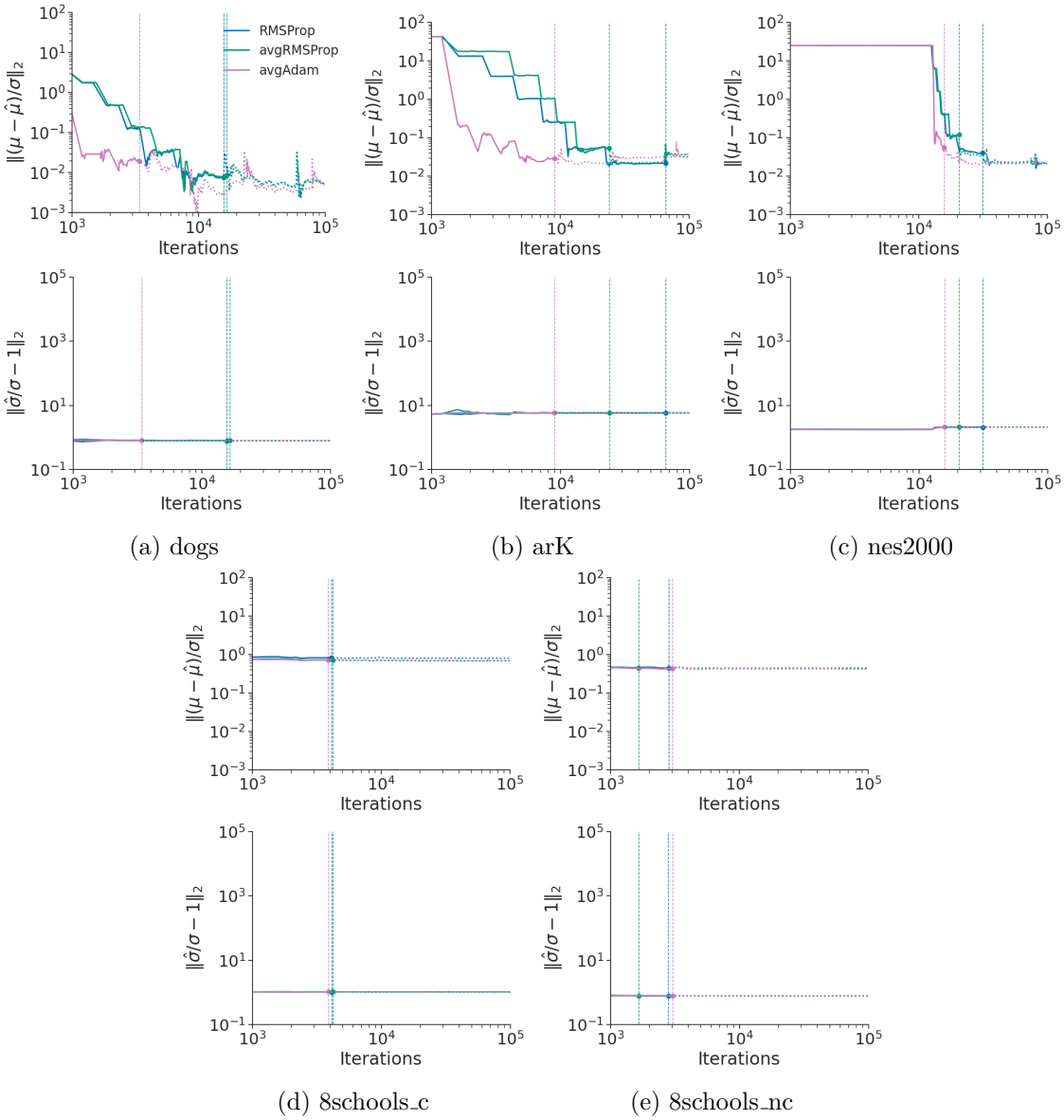
(a) uncorrelated $d = 100$

(b) uncorrelated $d = 500$

(c) uniform correlated $d = 100$

(d) banded correlated $d = 100$

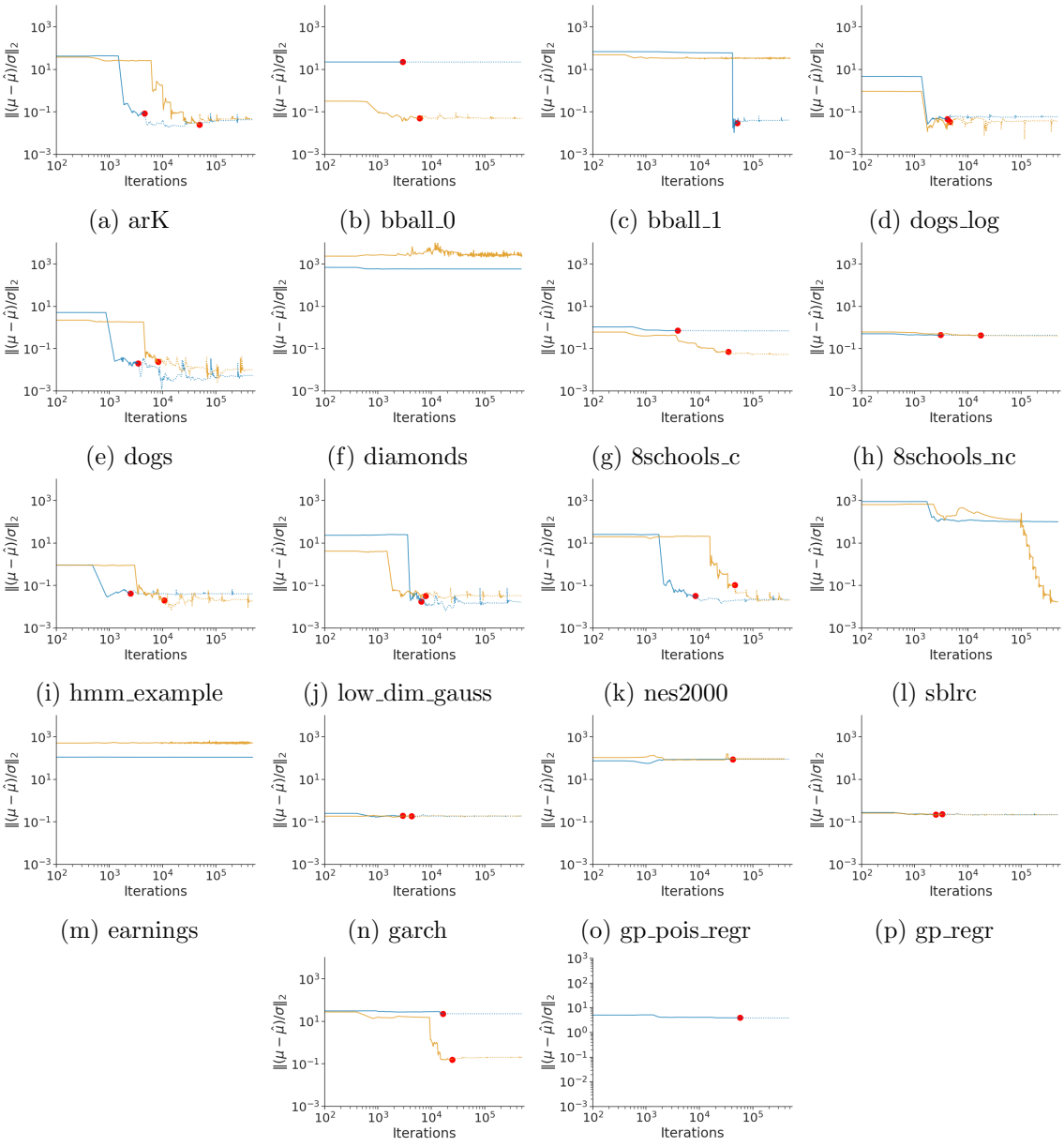
Figure C.10: Accuracy comparison of *RMSProp, avgRMSProp, avgAdam, SQN,* and *NGD* optimization methods in RABVI using Gaussian targets where accuracy is measured in terms of square root of symmetrized KL divergence between iterate average and optimal variational approximation. The vertical lines indicate the termination rule trigger points and the behavior of optimization methods at the trigger points showed in inset plots.



(a) dogs (b) arK (c) nes2000

(d) 8schools_c (e) 8schools_nc

Figure C.11: Accuracy comparison of *RMSProp, avgRMSProp,* and *avgAdam* optimization methods in RABVI using *posteriordb* datasets, where accuracy measured in terms of relative mean error (top) and relative standard deviation error (bottom). The vertical lines indicate the termination rule trigger points.



(a) arK (b) bball_0 (c) bball_1 (d) dogs_log

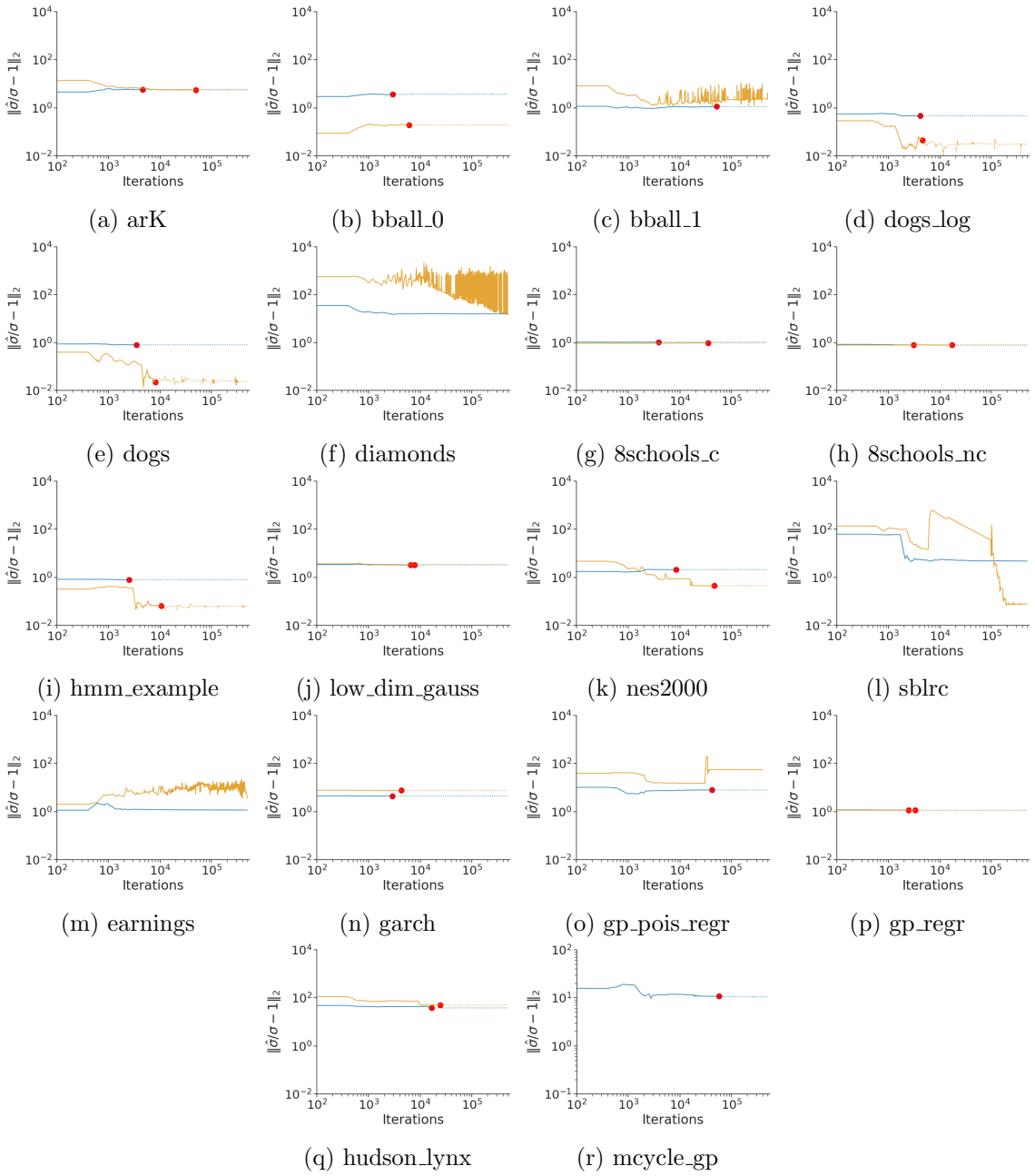
(e) dogs (f) diamonds (g) 8schools_c (h) 8schools_nc

(i) hmm_example (j) low_dim_gauss (k) nes2000 (l) sblrc

(m) earnings (n) garch (o) gp_pois_regr (p) gp_regr

(q) hudson_lynx (r) mcycle_gp

Figure C.12: Accuracy of mean-field (blue) and full-rank (orange) Gaussian family approximations for selected `posteriordb` data/models, where accuracy is measured in terms of relative mean error. The red dots indicate where the termination rule triggers



(a) arK (b) bball_0 (c) bball_1 (d) dogs_log

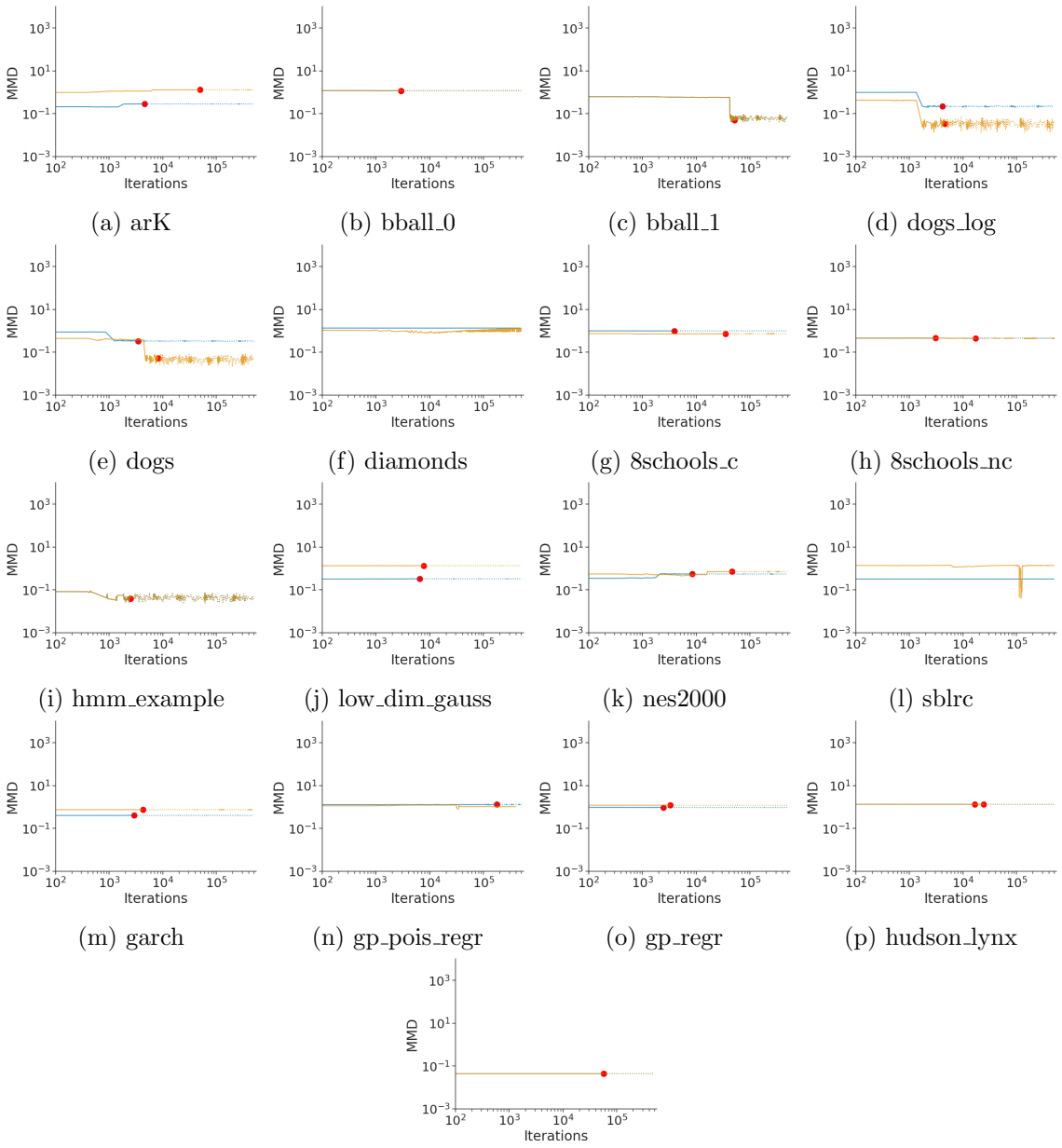
(e) dogs (f) diamonds (g) 8schools_c (h) 8schools_nc

(i) hmm_example (j) low_dim_gauss (k) nes2000 (l) sblrc

(m) earnings (n) garch (o) gp_pois_regr (p) gp_regr

(q) hudson_lynx (r) mcycle_gp

Figure C.13: Accuracy of mean-field (blue) and full-rank (orange) Gaussian family approximations for selected `posteriordb` data/models, where accuracy is measured in terms of relative standard deviation error. The red dots indicate where the termination rule triggers



(a) arK (b) bball_0 (c) bball_1 (d) dogs_log

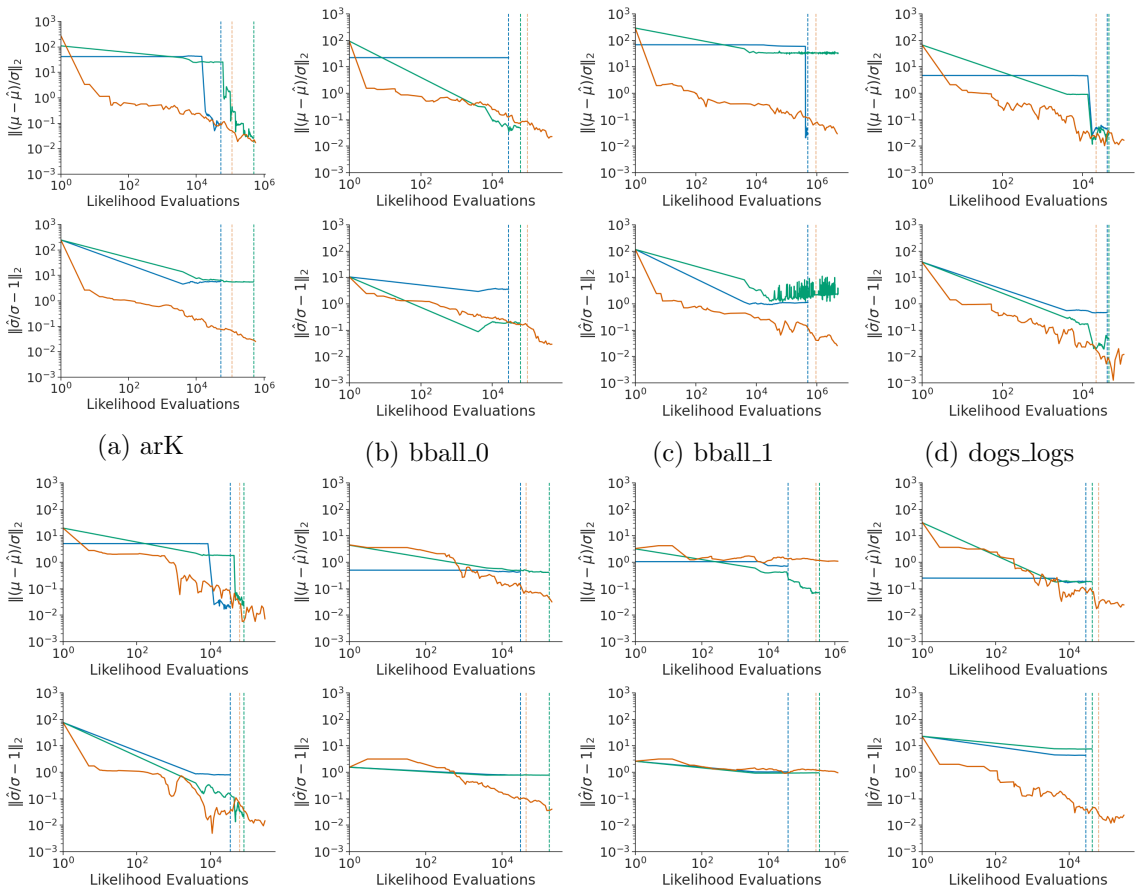
(e) dogs (f) diamonds (g) 8schools_c (h) 8schools_nc

(i) hmm_example (j) low_dim_gauss (k) nes2000 (l) sblrc

(m) garch (n) gp_pois_regr (o) gp_regr (p) hudson_lynx

(q) mcycle_gp

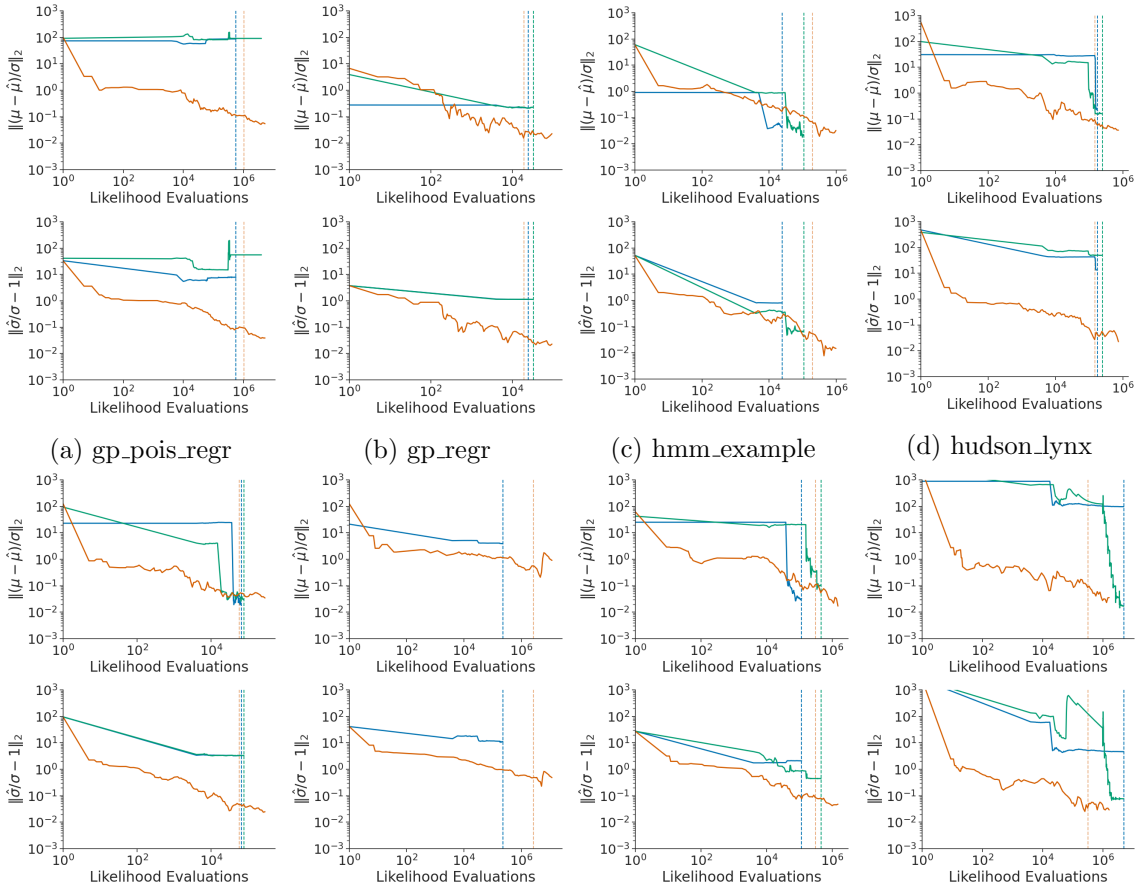
Figure C.14: Accuracy of mean-field (blue) and full-rank (orange) Gaussian family approximations for selected `posteriordb` data/models, where accuracy is measured in MMD. The red dots indicate where the termination rule triggers



(a) arK (b) bball_0 (c) bball_1 (d) dogs_logs

(e) dogs (f) 8schools_nc (g) 8schools_c (h) garch

Figure C.15: Results of RABVI with mean-field Gaussian (blue) and full-rank Gaussian (green) family comparison to dynamic HMC (orange) in terms of relative mean error (top) and relative standard error (bottom). Blue and green vertical lines show the termination rule triggering points in RABVI and orange vertical line shows the end of warm-up period in dynamic HMC.



(a) gp_pois_regr (b) gp_regr (c) hmm_example (d) hudson_lynx

(e) low_dim_gauss (f) mcycle_gp (g) nes2000 (h) sblrc

Figure C.16: Results of RABVI with mean-field Gaussian (blue) and full-rank Gaussian (green) family comparison to dynamic HMC (orange) in terms of relative mean error (top) and relative standard error (bottom). Blue and green vertical lines show the termination rule triggering points in RABVI and orange vertical line shows the end of warm-up period in dynamic HMC.

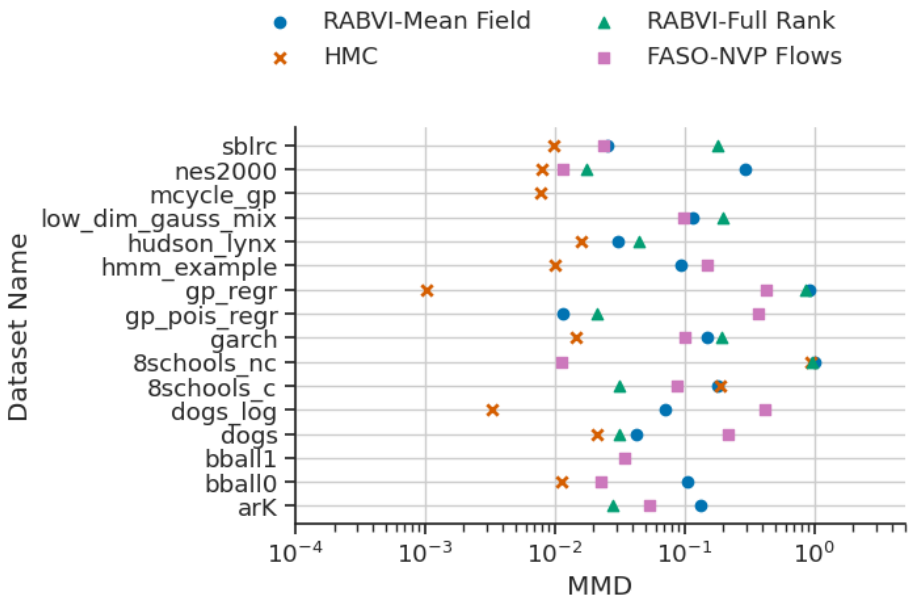


Figure C.17: Results of RABVI with mean-field Gaussian and full-rank Gaussian family and FASO with NVP flows comparison to dynamic HMC at the same computational cost (likelihood evaluations) in terms of MMD.

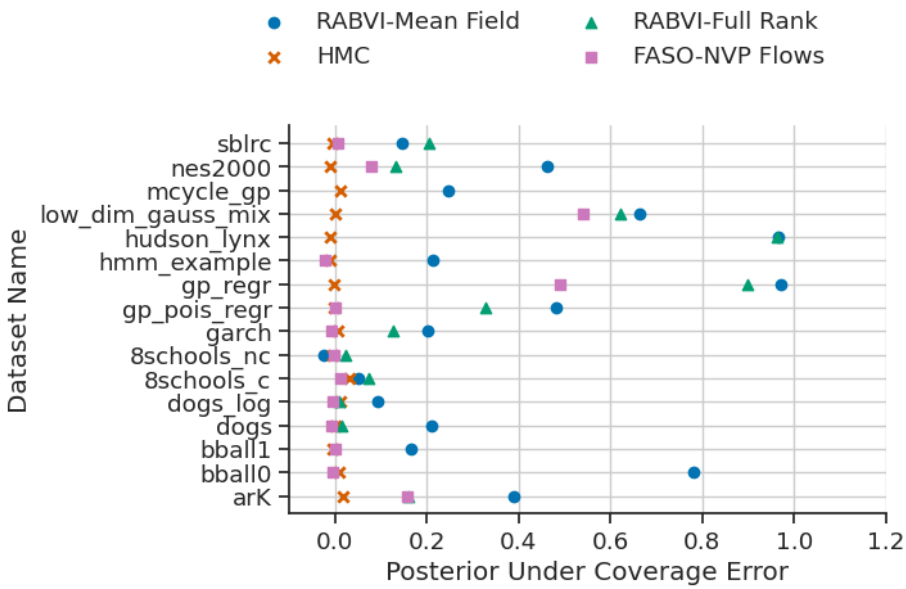
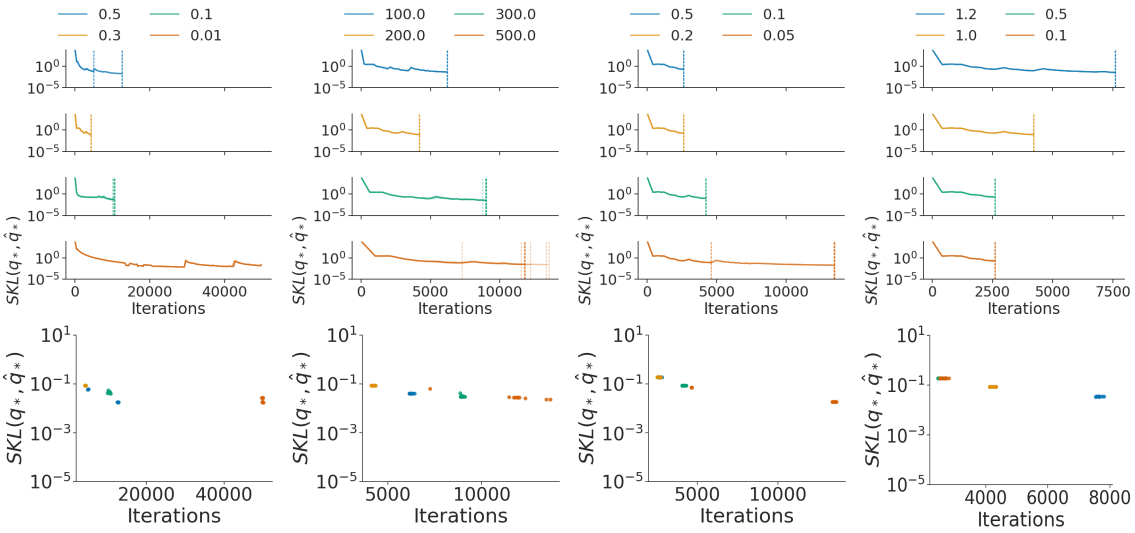
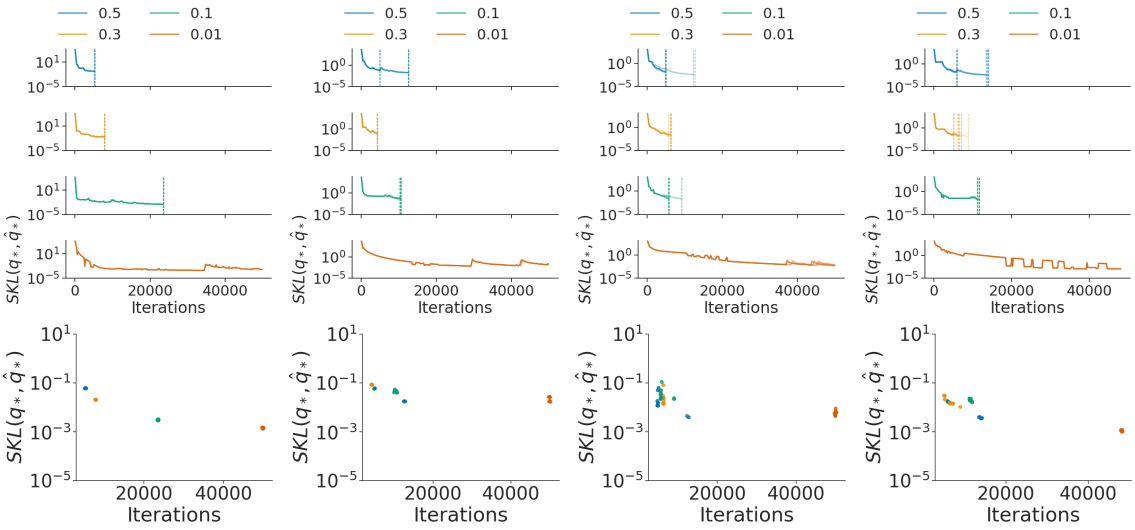


Figure C.18: Results of RABVI with mean-field Gaussian and full-rank Gaussian family and FASO with NVP flows comparison to dynamic HMC at the same computational cost (likelihood evaluations) in terms of posterior under coverage error of 95% quantiles.



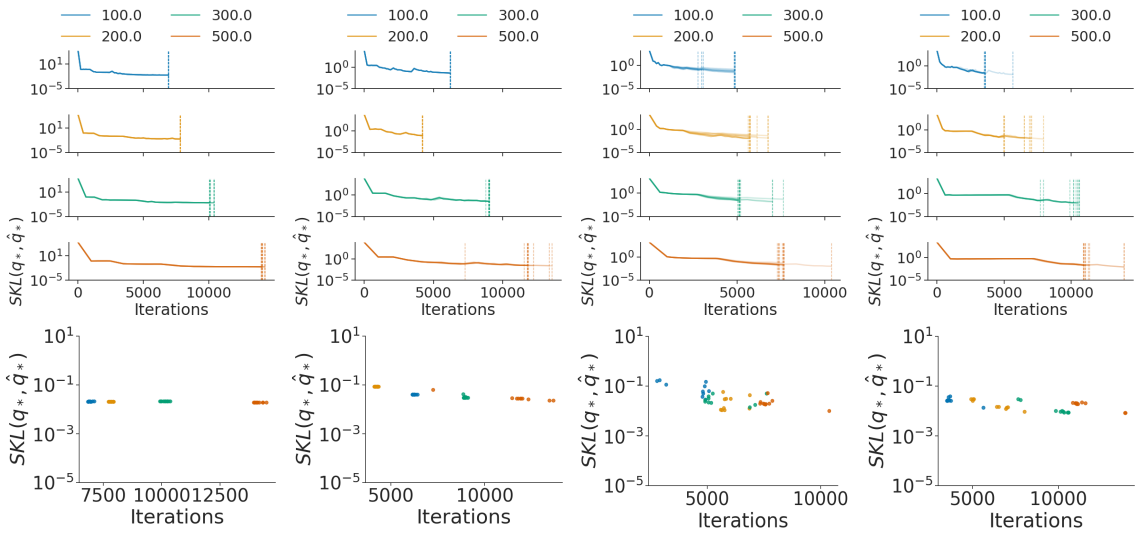
(a) γ_0 (b) W_{min} (c) ε_0 (d) τ

Figure C.19: Robustness to tuning parameters **(a)** initial learning rate γ_0, **(b)** minimum window size $W_{\min}$, **(c)** initial iterate average relative error threshold ε_0, and **(d)** inefficiency threshold τ. Results use Gaussian target $\mathcal{N}(0, V)$ with $d = 100$ and $V = I$ (identity covariance). **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



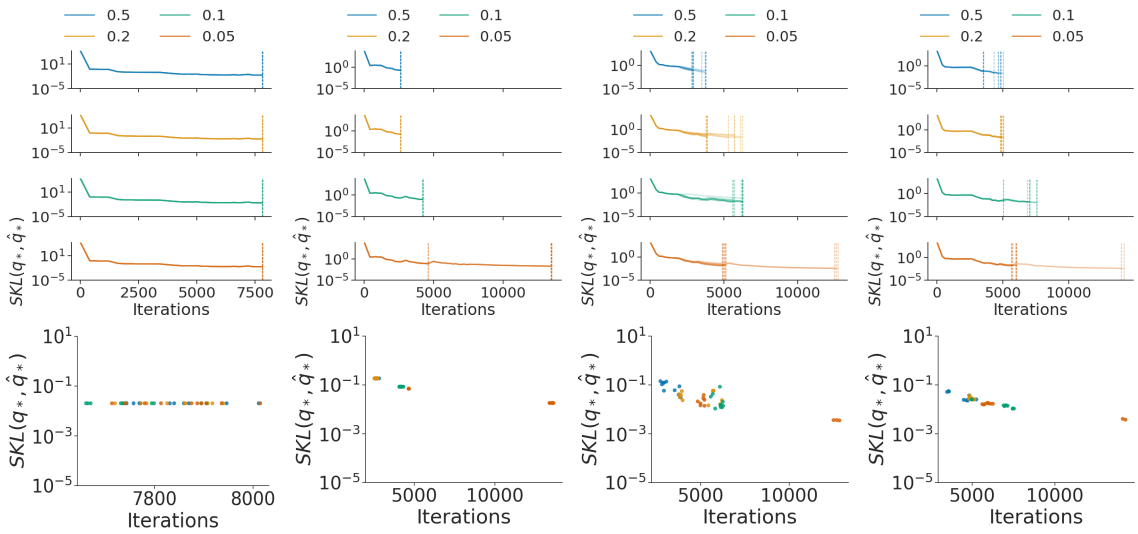
(a) uncorrelated $d = 100$ (b) uncorrelated $d = 500$ (c) uniform correlated $d = 100$ (d) banded correlated $d = 100$

Figure C.20: Robustness to initial learning rate γ_0 using Gaussian targets. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



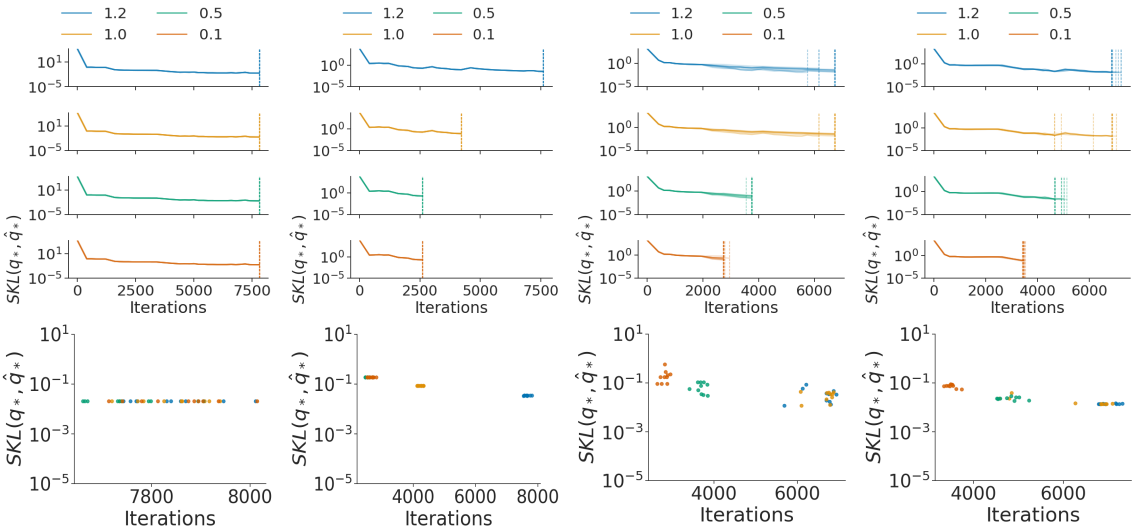
(a) uncorrelated $d = 100$ (b) uncorrelated $d = 500$ (c) uniform correlated $d = 100$ (d) banded correlated $d = 100$

Figure C.21: Robustness to minimum window size W_{min},, using Gaussian targets. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



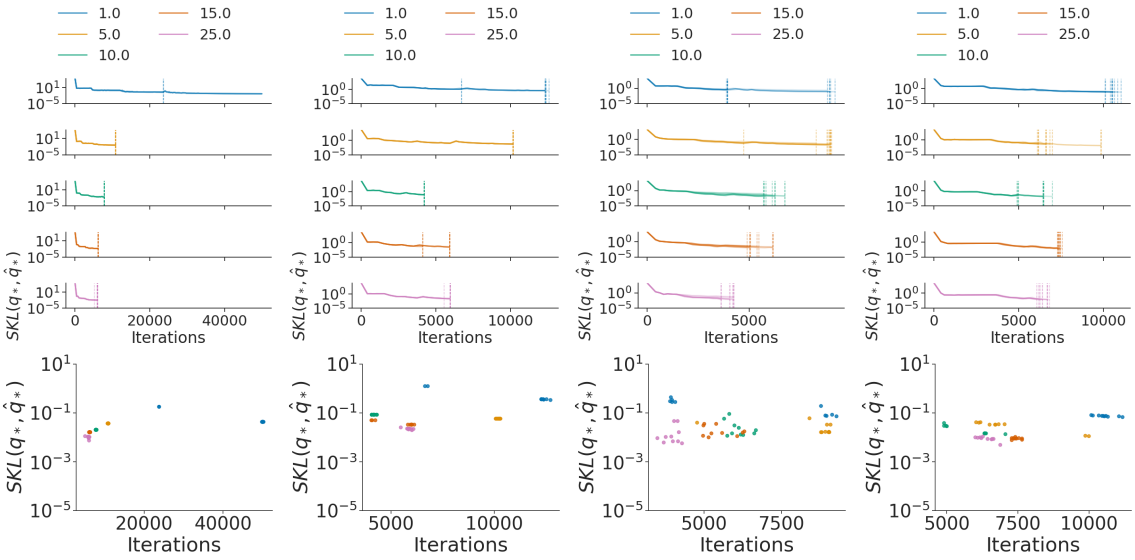
(a) uncorrelated $d = 100$ (b) uncorrelated $d = 500$ (c) uniform correlated $d = 100$ (d) banded correlated $d = 100$

Figure C.22: Robustness to initial iterate average relative error threshold ε using Gaussian targets. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



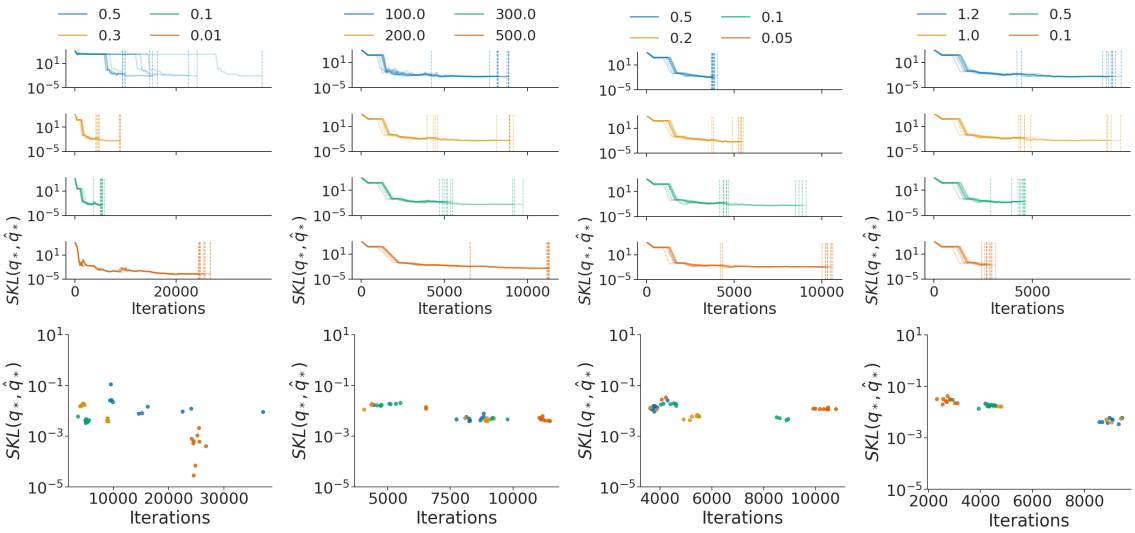
(a) uncorrelated $d = 100$ (b) uncorrelated $d = 500$ (c) uniform correlated $d = 100$ (d) banded correlated $d = 100$

Figure C.23: Robustness to inefficiency threshold τ using Gaussian targets. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



(a) uncorrelated $d = 100$ (b) uncorrelated $d = 500$ (c) uniform correlated $d = 100$ (d) banded correlated $d = 100$

Figure C.24: Robustness to Monte Carlo samples M using Gaussian targets. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The distinct lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.



(a) γ_0 (b) W_{min} (c) ε_0 (d) τ

Figure C.25: Robustness to tuning parameters (a) initial learning rate γ_0, (b) minimum window size $W_{\min}$, (c) initial iterate average relative error threshold ε_0, and (d) inefficiency threshold τ using *arK* dataset from `posteriordb` package. **(top)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation. The transparent lines represent repeated experiments and the vertical lines indicate the termination rule trigger points. **(bottom)** Iterations versus symmetrized KL divergence between iterate average and optimal variational approximation at the termination rule trigger point.

References

Abhinav Agrawal, Daniel Sheldon, and Justin Domke. Advances in Black-Box VI: Normalizing Flows, Importance Weighting, and Optimization. In *Advances in Neural Information Processing Systems*, 2020.

Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

F Bach and E Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*, pages 1–9, 2013.

Robert Bamler, Cheng Zhang, Manfred Opper, and Stephan Mandt. Perturbative black box variational inference. *Advances in Neural Information Processing Systems*, 30, 2017.

Kush Bhatia, Nikki Lijing Kuang, Yi-An Ma, and Yixin Wang. Statistical and computational trade-offs in variational inference: A case study in inferential model selection. *arXiv preprint arXiv:2207.11208*, 2022.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

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

François Bolley and C Villani. Weighted Csiszár-Kullback-Pinsker inequalities and applications to transportation inequalities. *Annales de la faculte des sciences de Toulouse*, 13 (3):331–352, 2005.

Ayman Boustati, Sattar Vakili, James Hensman, and ST John. Amortized variance reduction for doubly stochastic objective. In *Conference on Uncertainty in Artificial Intelligence*, pages 61–70. PMLR, 2020.

Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231 – 357, 2015. doi: 10.1561/2200000050.

Thang D Bui, Josiah Yan, and Richard E Turner. A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation. *Journal of Machine Learning Research*, 18:1–72, October 2017.

Y Burda, Roger B Grosse, and R Salakhutdinov. Importance Weighted Autoencoders. In *International Conference on Learning Representations*, 2016.

Jerry Chee and Ping Li. Understanding and Detecting Convergence for Stochastic Gradient Descent with Momentum. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 133–140. IEEE, 2020.

Jerry Chee and Panos Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Huiming Chen, Ho-Chun Wu, Shing-Chow Chan, and Wong-Hing Lam. A stochastic quasi-newton method for large-scale nonconvex optimization with applications. *IEEE transactions on neural networks and learning systems*, 31(11):4776–4790, 2019.

Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD, 2017. URL `https://openreview.net/forum?id=HyAddcLge`.

Julien Cornebise, E Moulines, and Jimmy Olsson. Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4): 461–480, August 2008.

Akash Kumar Dhaka, Alejandro Catalina, Michael Riis Andersen, Måns Magnusson, Jonathan H Huggins, and Aki Vehtari. Robust, Accurate Stochastic Optimization for Variational Inference. In *Advances in Neural Information Processing Systems*, volume 33, page 10961–10973, September 2020.

Akash Kumar Dhaka, Alejandro Catalina, Manushi Welandawe, Michael R Andersen, Jonathan Huggins, and Aki Vehtari. Challenges and opportunities in high dimensional variational inference. *Advances in Neural Information Processing Systems*, 34, 2021.

Adji B Dieng, Dustin Tran, Rajesh Ranganath, J. Paisley, and D. M. Blei. Variational Inference via χ Upper Bound Minimization. In *Advances in Neural Information Processing Systems*, 2017.

Aymeric Dieuleveut, Alain Durmus, and F Bach. Bridging the Gap between Constant Step Size Stochastic Gradient Descent and Markov Chains. *The Annals of Statistics*, 48(3): 1348–1382, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Justin Domke. Provable gradient variance guarantees for black-box variational inference. *Advances in Neural Information Processing Systems*, 32, 2019.

Justin Domke, Guillaume Garrigos, and Robert Gower. Provable convergence guarantees for black-box variational inference. *arXiv preprint arXiv:2306.03638*, 2023.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Alain Durmus and E Moulines. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, November 2019.

Alain Durmus, Szymon Majewski, and Blazej Miasojedow. Analysis of Langevin Monte Carlo via convex optimization. *Journal of Machine Learning Research*, 20:1–46, 2019.

Andreas Eberle and Mateusz B Majka. Quantitative contraction rates for Markov chains on general state spaces. *Electronic Journal of Probability*, 24(0):1–36, 2019.

Yarin Gal and Z. Ghahramani. Dropout as a Bayesian Approximation - Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, 2016.

Tomas Geffner and Justin Domke. Using large ensembles of control variates for variational inference. *Advances in Neural Information Processing Systems*, 31, 2018.

Tomas Geffner and Justin Domke. Approximation based variance reduction for reparameterization gradients. *Advances in Neural Information Processing Systems*, 33:2397–2407, 2020.

Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–511, 1992.

Andrew Gelman, John Carlin, Hal Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, third edition, 2013.

C J Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992.

Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. Understanding the Role of Momentum in Stochastic Gradient Methods. In *Advances in Neural Information Processing Systems*, pages 9633–9643, 2019.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Thang D Bui, Daniel Hernández-Lobato, and Richard E Turner. Black-Box Alpha Divergence Minimization. In *International Conference on Machine Learning*, 2016.

G. E. Hinton and Tijmen Tieleman. Lecture 6.5 – Rmsprop: Divide the gradient by a running average of its recent magnitude. In *Coursera: Neural networks for machine learning*, 2012.

Jonathan H Huggins, Mikolaj Kasprzak, Trevor Campbell, and T. Broderick. Validated Variational Inference via Practical Posterior Error Bounds. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Matthew J Johnson, D Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and R P Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

Aldéric Joulin and Yann Ollivier. Curvature, concentration and error estimates for Markov chain Monte Carlo. *The Annals of Probability*, 38(6):2418–2442, 2010.

Mohammad Emtiyaz Khan and Wu Lin. Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Kyurae Kim, Jisu Oh, Kaiwen Wu, Yian Ma, and Jacob R. Gardner. On the convergence of black-box variational inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dHQ2av9NzO.

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

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and D. M. Blei. Automatic Variational Inference in Stan. In *Advances in Neural Information Processing Systems*, June 2015.

Hunter Lang, Pengchuan Zhang, and Lin Xiao. Using Statistics to Automate Stochastic Optimization. *Advances in neural information processing systems*, 32, 2019.

Yingzhen Li and Richard E Turner. Rényi Divergence Variational Inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

Sifan Liu and Art B Owen. Quasi-monte carlo quasi-newton in variational bayes. *Journal of Machine Learning Research*, 22(243):1–23, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A Simple Baseline for Bayesian Uncertainty in Deep Learning. In *Advances in Neural Information Processing Systems*, 2019.

Neal Madras and Deniz Sezer. Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances. *Bernoulli*, 16(3):882–908, August 2010.

Andrew Miller, Nick Foti, Alexander D'Amour, and Ryan P Adams. Reducing reparameterization gradient variance. *Advances in Neural Information Processing Systems*, 30, 2017.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo Gradient Estimation in Machine Learning. *Journal of Machine Learning Research*, 21(132): 1–62, 2020.

Mahesh Chandra Mukkamala and Matthias Hein. Variants of RMSProp and Adagrad with Logarithmic Regret Bounds. In *International Conference on Machine Learning*, 2017.

K Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

John William Paisley, David M. Blei, and Michael I. Jordan. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012. URL `https://api.semanticscholar.org/CorpusID:1758804`.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Scott Pesme, Aymeric Dieuleveut, and Nicolas Flammarion. On Convergence-Diagnostic based Step Sizes for Stochastic Gradient Descent. In *International Conference on Machine Learning*, 2020.

Georg Ch Pflug. Non-asymptotic confidence bounds for stochastic approximation algorithms with constant step size. *Monatshefte für Mathematik*, 110(3):297–314, September 1990.

B T Polyak and A B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal of Control and Optimization*, 30(4):838–855, July 1992.

Rajesh Ranganath, Sean Gerrish, and D. M. Blei. Black Box Variational Inference. In *International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*, 2014.

C P Robert. *The Bayesian Choice*. Springer, New York, NY, 2nd edition, 2007.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. *Advances in Neural Information Processing Systems*, 30, 2017.

Daniel Rudolf and Nikolaus Schweizer. Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli*, 4A:2610–2639, 2018.

Francisco R Ruiz, Titsias RC AUEB, David Blei, et al. The generalized reparameterization gradient. *Advances in neural information processing systems*, 29, 2016.

D Ruppert. Efficient estimations from a slowly convergent Robbins–Monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

Yunus Saatchi and Andrew Gordon Wilson. Bayesian GAN. In *Advances in Neural Information Processing Systems*, 2017.

Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *ArXiv*, 2013.

Tim Salimans, Diederik P Kingma, and Max Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In *International Conference on Machine Learning*, 2015.

Stan Development Team. Stan modeling language users guide and reference manual, 2020. URL `https://mc-stan.org`.

Michalis K Titsias and Miguel Lázaro-Gredilla. Doubly Stochastic Variational Bayes for non-Conjugate Inference. In *International Conference on Machine Learning*, 2014.

Dootika Vats and Christina Knudson. Revisiting the gelman-rubin diagnostic. *Statistical Science*, 36(4):518–529, 2021.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC. *Bayesian Analysis*, 16(2):667–718, 2021.

C Villani. *Optimal transport: old and new*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009.

Sebastian J Vollmer, Konstantinos C Zygalakis, and Y W Teh. (Non-) asymptotic properties of Stochastic Gradient Langevin Dynamics. *Journal of Machine Learning Research*, 17 (159):1–48, 2016.

M. J. Wainwright and M I Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Neng Wan, Dapeng Li, and Naira Hovakimyan. f-Divergence Variational Inference. *Advances in neural information processing systems*, 33:17370–17379, 2020.

Dilin Wang, Hao Liu, and Qiang Liu. Variational Inference with Tail-adaptive f-Divergence. In *Advances in Neural Information Processing Systems*, 2018.

Xi Wang, Tomas Geffner, and Justin Domke. A dual control variate for accelerated black-box variational inference. In *Fifth Symposium on Advances in Approximate Bayesian Inference*, 2023a. URL `https://openreview.net/forum?id=3c5YBeZ06X`.

Xi Wang, Tomas Geffner, and Justin Domke. Joint control variate for faster black-box variational inference, 2023b.

Yu Wang, Mikolaj Kasprzak, and Jonathan H Huggins. A targeted accuracy diagnostic for variational approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 8351–8372. PMLR, 2023c.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. In *International Conference on Learning Representations*, 2019.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but Did It Work?: Evaluating Variational Inference. In *International Conference on Machine Learning*, volume 80 of *PMLR*, pages 5577–5586, 2018.

Pengchuan Zhang, Hunter Lang, Qiang Liu, and Lin Xiao. Statistical Adaptive Stochastic Gradient Methods. *arXiv.org*, arXiv:2002.10597 [stat.ML], February 2020.