Batch and match: black-box variational inference with a score-based divergence

Diana Cai [1] **Chirag Modi** [1 2] **Loucas Pillaud-Vivien** [1 3] **Charles C. Margossian** [1] **Robert M. Gower** [1]
David M. Blei [4] **Lawrence K. Saul** [1]

Abstract

Most leading implementations of black-box variational inference (BBVI) are based on optimizing a stochastic evidence lower bound (ELBO). But such approaches to BBVI often converge slowly due to the high variance of their gradient estimates and their sensitivity to hyperparameters. In this work, we propose *batch and match* (BaM), an alternative approach to BBVI based on a score-based divergence. Notably, this score-based divergence can be optimized by a closed-form proximal update for Gaussian variational families with full covariance matrices. We analyze the convergence of BaM when the target distribution is Gaussian, and we prove that in the limit of infinite batch size the variational parameter updates converge exponentially quickly to the target mean and covariance. We also evaluate the performance of BaM on Gaussian and non-Gaussian target distributions that arise from posterior inference in hierarchical and deep generative models. In these experiments, we find that BaM typically converges in fewer (and sometimes significantly fewer) gradient evaluations than leading implementations of BBVI based on ELBO maximization.

1. Introduction

Probabilistic modeling plays a fundamental role in many problems of inference and decision-making, but it can be challenging to develop accurate probabilistic models that remain computationally tractable. In typical applications, the goal is to estimate a target distribution that cannot be evaluated or sampled from exactly, but where an unnormalized form is available. A canonical situation is applied Bayesian statistics, where the target is a posterior distribution of latent variables given observations, but where only the model's joint distribution is available in closed form. Variational inference (VI) has emerged as a leading method for fast approximate inference (Jordan et al., 1999; Wainwright et al., 2008; Blei et al., 2017). The idea behind VI is to posit a parameterized family of approximating distributions, and then to find the member of that family which is closest to the target distribution.

Recently, VI methods have become increasingly "black box," in that they only require calculation of the log of the unnormalized target and (for some algorithms) its gradients (Ranganath et al., 2014; Kingma & Welling, 2014; Archer et al., 2015; Ryder et al., 2018; Locatello et al., 2018; Burroni et al., 2023; Kim et al., 2023; Domke, 2019; Welandawe et al., 2022; Domke et al., 2023; Modi et al., 2023; Giordano et al., 2024). Further applications have built on advances in automatic differentiation, and now black-box variational inference (BBVI) is widely deployed in robust software packages for probabilistic programming (Salvatier et al., 2016; Kucukelbir et al., 2017; Bingham et al., 2019).

In general, the ingredients of a BBVI strategy are the form of the approximating family, the divergence to be minimized, and the optimization algorithm to minimize it. Most BBVI algorithms work with a factorized (or mean-field) family, and minimize the reverse Kullback-Leibler (KL) divergence via stochastic gradient descent (SGD). But this approach has its drawbacks. The optimizations can be plagued by high-variance gradients and sensitivity to hyperparameters of the learning algorithms (Dhaka et al., 2020; 2021). These issues are further exacerbated in high-dimensional problems and when using richer variational families that model the correlations between different latent variables. There has been recent work on BBVI which avoids SGD for Gaussian variational families (Modi et al., 2023), but this approach does not minimize an explicit divergence and requires additional heuristics to converge for non-Gaussian targets.

In this paper, we develop a new approach to BBVI. It is based on a different divergence, accommodates expressive variational families, and does not rely on SGD for optimization. In particular, we introduce a novel *score-based divergence* that measures the agreement of the scores, or

[1]Center for Computational Mathematics, Flatiron Institute [2]Center for Computational Astrophysics, Flatiron Institute [3]CERMICS Laboratory, Ecole des Ponts ParisTech [4]Department of Statistics, Department of Computer Science, Columbia University. Correspondence to: Diana Cai <dcai@flatironinstitute.org>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

gradients of the log densities, of the target and variational distributions. This divergence can be estimated for unnormalized target distributions, thus making it a natural choice for BBVI. We study the score-based divergence for Gaussian variational families with full covariance, rather than the factorized family. We also develop an efficient stochastic proximal point algorithm, with closed-form updates, to optimize this divergence.

Our algorithm is called *batch and match* (BaM), and it alternates between two types of steps. In the "batch" step, we draw a batch of samples from the current approximation to the target and use those samples to estimate the divergence; in the "match" step, we estimate a new variational approximation by matching the scores at these samples. By iterating these steps, BaM finds a variational distribution that is close in score-based divergence to the target.

Theoretically, we analyze the convergence of BaM when the target itself is Gaussian. In the limit of an infinite batch size, we prove that the variational parameters converge exponentially quickly to the target mean and covariance at a rate controlled by the quality of initialization and the amount of regularization. Notably, this convergence result holds for any amount of regularization; this stability to the "learning rate" parameter is characteristic of proximal algorithms, which are often less brittle than SGD (Asi & Duchi, 2019).

Empirically, we evaluate BaM on a variety of Gaussian and non-Gaussian target distributions, including a test suite of Bayesian hierarchical models and deep generative models. On these same problems, we also compare BaM to a leading implementation of BBVI based on ELBO maximization (Kucukelbir et al., 2017) and a recently proposed algorithm for Gaussian score matching (Modi et al., 2023). By and large, we find that BaM converges faster and to more accurate solutions.

In what follows, we begin by reviewing BBVI and then developing a score-based divergence for BBVI with several important properties (Section 2). Next, we propose BaM, an iterative algorithm for score-based Gaussian variational inference, and we study its rate of convergence (Section 3). We then present a discussion of related methods in the literature (Section 4). Finally, we conclude with a series of empirical studies on a variety of synthetic and real-data target distributions (Section 5). A Python implementation of BaM is available at github.com/modichirag/GSM-VI/.

2. BBVI with the score-based divergence

VI was developed as a way to estimate an unknown *target* distribution with density p; here we assume that the target is a distribution on $\mathbb{R}^D$. The target is estimated by first positing a *variational family* of distributions $\mathcal{Q}$, then finding the particular $q \in \mathcal{Q}$ that minimizes an objective $\mathscr{L}(q)$ measuring the difference between p and q.

2.1. From VI to BBVI to score-based BBVI

In the classical formulation of VI, the objective $\mathscr{L}(q)$ is the (reverse) Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(q;p) := \int \log\left(\frac{q(z)}{p(z)}\right) q(z)\, dz. \tag{1}$$

For some models the derivatives of $\mathrm{KL}(q;p)$ can be exactly evaluated, but for many others they cannot. In this case a further approximation is needed. This more challenging situation is the typical setting for BBVI.

In BBVI, it is assumed that (a) the target density p cannot be evaluated pointwise or sampled from exactly, but that (b) an unnormalized target density is available. BBVI algorithms use stochastic gradient descent to minimize the KL divergence, or equivalently, to maximize the evidence lower bound (ELBO). The necessary gradients in this case can be estimated with access to the unnormalized target density. But in practice this objective is difficult to optimize: the optimization can converge slowly due to noisy gradients, and it can be sensitive to the choice of learning rates.

In this work, we will also assume additionally that (c) the log target density is differentiable, and its derivatives can be efficiently evaluated. We define the target density's *score function* $s : \mathbb{R}^D \to \mathbb{R}^D$ as

$$s(z) := \nabla_z \log p(z).$$

It is often possible to compute these scores even when p is intractable because they only depend on the logarithm of the unormalized target density. In what follows, we introduce the score-based divergence and study its properties; in Section 3, we will then propose a BBVI algorithm based on this score-based divergence.

Notation. For $\Sigma \in \mathbb{R}^{D\times D}$, let $\Sigma \succ 0$ denote that Σ is positive definite and $\Sigma \succeq 0$ denote that Σ is positive semi-definite. Define the set of symmetric, positive definite matrices as $\mathbb{S}^D_{++} := \{\Sigma \in \mathbb{R}^{D\times D} : \Sigma = \Sigma^\top, \Sigma \succ 0\}$. Let $\mathrm{tr}(\Sigma) := \sum_{d=1}^D \Sigma_{dd}$ denote the trace of Σ and let $I \in \mathbb{R}^{D\times D}$ denote the identity matrix. We primarily consider two norms throughout the paper: first, given $z \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D\times D}$, we define the Σ-weighted vector norm, $\|z\|_\Sigma := \sqrt{z^\top \Sigma z}$, and second, given $\Sigma \in \mathbb{R}^{D\times D}$, we define the matrix norm $\|\Sigma\|$ to be the spectral norm.

2.2. The score-based divergence

We now introduce the score-based divergence, which will be the basis for a BBVI objective. Here we focus on a Gaussian variational family, i.e.,

$$\mathcal{Q} = \{\mathcal{N}(\mu, \Sigma) : \mu \in \mathbb{R}^D, \Sigma \in \mathbb{S}^D_{++}\},$$

but we generalize the score-based divergence to non-Gaussian distributions in Appendix A.

The *score-based divergence* between densities $q \in \mathcal{Q}$ and p on $\mathbb{R}^D$ is defined as

$$\mathscr{D}(q;p) := \int \left\| \nabla_z \log \left(\frac{q(z)}{p(z)} \right) \right\|^2_{\mathrm{Cov}(q)} q(z)\, dz, \quad (2)$$

where $\mathrm{Cov}(q) \in \mathbb{S}^D_{++}$ is the covariance matrix of the variational density q.

Importantly, the score-based divergence can be evaluated when p is only known up to a normalization constant, as it only depends on the target density through the score $\nabla \log p$. Thus, not only can this divergence be used as a VI objective, but it can also be used for goodness-of-fit evaluations, unlike the KL divergence.

The divergence in eq. (2) is well-defined under mild conditions on p and q (see Appendix A), and it enjoys two important properties:

Property 1 (Non-negativity & equality): $\mathscr{D}(q;p) \geq 0$ with $\mathscr{D}(q;p) = 0$ iff $p = q$.

Property 2 (Affine invariance): Let $h : \mathbb{R}^D \to \mathbb{R}^D$ be an affine transformation, and consider the induced densities $\tilde{q}(h(z)) = q(z)|\mathcal{J}(z)|^{-1}$ and $\tilde{p}(h(z)) = p(z)|\mathcal{J}(z)|^{-1}$, where $\mathcal{J}$ is the determinant of the Jacobian of h. Then $\mathscr{D}(q;p) = \mathscr{D}(\tilde{q};\tilde{p})$.

We note that these properties are also satisfied by the KL divergence (Qiao & Minematsu, 2010). The first property shows that $\mathscr{D}(q;p)$ is a proper divergence measuring the agreement between p and q. The second property states that the score-based divergence $\mathscr{D}(q,p)$ is invariant under affine transformations; this property is desirable to maintain a consistent measure of similarity under coordinate transformations of the input. This property depends crucially on the weighted vector norm, mediated by $\mathrm{Cov}(q)$, in the divergence of eq. (2).

There are several related divergences in the research literature. A generalization of the score-based divergence is the weighted Fisher divergence (Barp et al., 2019), given by $\mathbb{E}_q[\|\nabla \log(q/p)\|^2_M]$, where $M \in \mathbb{R}^{D\times D}$; the score-based divergence is recovered by the choice $M = \mathrm{Cov}(q)$. A special case of the score-based divergence is the Fisher divergence (Hyvärinen, 2005) given by $\mathbb{E}_q[\|\nabla \log(q/p)\|^2_I]$, but this divergence is not affine invariant. (See the proof of Theorem A.4 for further discussion.)

3. Score-based Gaussian variational inference

The score-based divergence has many favorable properties for VI. We now show that this divergence can also be efficiently optimized by an iterative black-box algorithm.

3.1. Algorithm

Our goal is to find some Gaussian distribution $q^* \in \mathcal{Q}$ that minimizes $\mathscr{D}(q;p)$. Without additional assumptions on the target p, the score-based divergence $\mathscr{D}(q;p)$ is not analytically tractable. So instead we consider a Monte Carlo estimate of $\mathscr{D}(q;p)$: given samples $z_1, \ldots, z_B \sim q$, we construct the approximation

$$\mathscr{D}(q;p) \approx \frac{1}{B} \sum_{b=1}^{B} \left\| \nabla_z \log \left(\frac{q(z_b)}{p(z_b)} \right) \right\|^2_{\mathrm{Cov}(q)}. \quad (3)$$

This estimator is unbiased, but it does not lend itself to optimization: we cannot simultaneously sample from q while also optimizing over the family $\mathcal{Q}$ to which it belongs. There is a generic solution to the above problem: the so-called "reparameterization trick" (e.g., Kucukelbir et al. (2017)) decouples the sampling distribution and optimization variable. But this approach leads to a gradient-based algorithm that does not fully capitalize on the structure of the Gaussian variational family.

In this paper we take a different approach, one that does capitalize on this structure. Specifically, we take an iterative approach whose goal is to produce a sequence of distributions $\{q_t\}_{t=0}^{\infty}$ that converges to q^*. At a high level, the approach alternates between two steps—one that constructs a biased estimate of $\mathscr{D}(q;p)$, and another that updates q based on this biased estimate, but not too aggressively (so as to minimize the effect of the bias). Specifically, at the tth iteration, we first estimate $\mathscr{D}(q;p)$ with samples from q_t: i.e., given $z_1, \ldots, z_B \sim q_t$, we compute

$$\widehat{\mathscr{D}}_{q_t}(q;p) := \frac{1}{B} \sum_{b=1}^{B} \left\| \nabla_z \log \left(\frac{q(z_b)}{p(z_b)} \right) \right\|^2_{\mathrm{Cov}(q)}. \quad (4)$$

We call eq. (4) the *batch* step because it estimates $\mathscr{D}(q,p)$ from the batch of samples $z_1, \ldots, z_B \sim q_t$.

The batch step of the algorithm relies on stochastic sampling, but it alternates with a deterministic step that updates q by minimizing the empirical score-based divergence $\widehat{\mathscr{D}}_{q_t}(q;p)$ in eq. (4). Importantly, this minimization is subject to a regularizer: we penalize large differences between q_t and q_{t+1} by their KL divergence. Intuitively, when q remains close to q_t, then $\widehat{\mathscr{D}}_{q_t}(q;p)$ in eq. (4) remains a good approximation to the unbiased estimate $\widehat{\mathscr{D}}_q(q;p)$ in eq. (3). With this in mind, we compute q_{t+1} by minimizing the *regularized* objective function

$$\mathscr{L}^{\mathrm{BaM}}(q) := \widehat{\mathscr{D}}_{q_t}(q;p) + \tfrac{2}{\lambda_t}\, \mathrm{KL}(q_t;q), \quad (5)$$

where $q \in \mathcal{Q}$ and $\lambda_t > 0$ is the inverse regularization parameter. When λ_t is small, the regularizer is large, encouraging the next iterate q_{t+1} to remain close to q_t; thus λ_t can also be viewed as a learning rate.

The objective function in eq. (5) has the important property that its global minimum can be computed analytically in closed form. In particular, we can optimize eq. (5) without recourse to gradient-based methods that are derived from a linearization around q_t. We refer to the minimization of $\mathscr{L}^{\text{BaM}}(q)$ in eq. (5) as the *match* step because the updated distribution q_{t+1} always matches the scores at $z_1, \ldots, z_B$ better than the current one q_t.

Combining these two steps, we arrive at the *batch and match* (BaM) algorithm for BBVI with a score-based divergence. The intuition behind this iterative approach will be formally justified in Section 3.2 by a proof of convergence. We now discuss each step of the algorithm in greater detail.

Batch Step. This step begins by sampling $z_1, z_2, \ldots, z_B \sim q_t$ and computing the scores $g_b = \nabla \log p(z_b)$ at each sample. It then calculates the means and covariances (over the batch) of these quantities; we denote these statistics by

$$\overline{z} = \frac{1}{B}\sum_{b=1}^{B} z_b, \quad C = \frac{1}{B}\sum_{b=1}^{B}(z_b - \overline{z})(z_b - \overline{z})^\top \tag{6}$$

$$\overline{g} = \frac{1}{B}\sum_{b=1}^{B} g_b, \quad \Gamma = \frac{1}{B}\sum_{b=1}^{B}(g_b - \overline{g})(g_b - \overline{g})^\top, \tag{7}$$

where $\overline{z}, \overline{g} \in \mathbb{R}^D$ are the means, respectively, of the samples and the scores, and $C, \Gamma \in \mathbb{R}^{D\times D}$ are their covariances. In Appendix C, we show that the empirical score-based divergence $\widehat{\mathscr{D}}_{q_t}(q;p)$ in eq. (4) can be written in terms of these statistics as

$$\widehat{\mathscr{D}}_{q_t}(q;p) = \operatorname{tr}(\Gamma\Sigma) + \operatorname{tr}(C\Sigma^{-1}) + \left\|\mu - \overline{z} - \Sigma\overline{g}\right\|^2_{\Sigma^{-1}} + \text{const.},$$

where for clarity we have suppressed additive constants that do not depend on the mean μ or covariance Σ of q. This calculation completes the batch step of BaM.

Match Step. The match step of BaM updates the variational approximation q by setting

$$q_{t+1} = \arg\min_{q\in\mathcal{Q}} \mathscr{L}^{\text{BaM}}(q), \tag{8}$$

where $\mathscr{L}^{\text{BaM}}(q)$ is given by eq. (5). This optimization can be solved in closed form; that is, we can analytically calculate the variational mean μ_{t+1} and covariance Σ_{t+1} that minimize $\mathscr{L}^{\text{BaM}}(q)$.

The details of this calculation are given in Appendix C. There we show that the updated covariance Σ_{t+1} satisfies a quadratic matrix equation,

$$\Sigma_{t+1} U \Sigma_{t+1} + \Sigma_{t+1} = V, \tag{9}$$

where the matrices U and V in this expression are positive semidefinite and determined by statistics from the batch step

Algorithm 1 Batch and match VI

1: **Input:** Iterations T, batch size B, inverse regularization $\lambda_t > 0$, target score function $s : \mathbb{R}^D \to \mathbb{R}^D$, initial variational mean $\mu_0 \in \mathbb{R}^D$ and covariance $\Sigma_0 \in \mathbb{S}^D_{++}$
2: **for** $t = 0, \ldots, T-1$ **do**
3: Sample batch $z_b \sim \mathcal{N}(\mu_t, \Sigma_t)$ for $b = 1, \ldots, B$
4: Evaluate scores $g_b = s(z_b)$ for $b = 1, \ldots, B$
5: Compute statistics $\overline{z}, \overline{g} \in \mathbb{R}^D$ and $\Gamma, C \in \mathbb{R}^{D\times D}$

$$\overline{z} = \tfrac{1}{B}\sum_{b=1}^{B} z_b, \qquad C = \tfrac{1}{B}\sum_{b=1}^{B}(z_b - \overline{z})(z_b - \overline{z})^\top$$

$$\overline{g} = \tfrac{1}{B}\sum_{b=1}^{B} g_b, \qquad \Gamma = \tfrac{1}{B}\sum_{b=1}^{B}(g_b - \overline{g})(g_b - \overline{g})^\top$$

6: Compute matrices U and V needed to solve the quadratic matrix equation $\Sigma U \Sigma + \Sigma = V$

$$U = \lambda_t \Gamma + \tfrac{\lambda_t}{1+\lambda_t}\,\overline{g}\,\overline{g}^\top$$

$$V = \Sigma_t + \lambda_t C + \tfrac{\lambda_t}{1+\lambda_t}(\mu_t - \overline{z})(\mu_t - \overline{z})^\top$$

7: Update variational parameters

$$\Sigma_{t+1} = 2\,V\Big(I + (I + 4\,UV)^{\frac{1}{2}}\Big)^{-1}$$

$$\mu_{t+1} = \tfrac{1}{1+\lambda_t}\mu_t + \tfrac{\lambda_t}{1+\lambda_t}\left(\Sigma_{t+1}\,\overline{g} + \overline{z}\right)$$

8: **end for**
9: **Output:** variational parameters μ_T, Σ_T

of BaM. In particular, these matrices are given by

$$U = \lambda_t \Gamma + \tfrac{\lambda_t}{1+\lambda_t}\,\overline{g}\,\overline{g}^\top \tag{10}$$

$$V = \Sigma_t + \lambda_t C + \tfrac{\lambda_t}{1+\lambda_t}(\mu_t - \overline{z})(\mu_t - \overline{z})^\top. \tag{11}$$

The quadratic matrix equation in eq. (9) has a symmetric and positive-definite solution (see Appendix B), and it is given by

$$\Sigma_{t+1} = 2\,V\Big(I + (I + 4\,UV)^{\frac{1}{2}}\Big)^{-1}. \tag{12}$$

The solution in eq. (12) is the BaM update for the variational covariance. The update for the variational mean is given by

$$\mu_{t+1} = \tfrac{1}{1+\lambda_t}\mu_t + \tfrac{\lambda_t}{1+\lambda_t}\left(\Sigma_{t+1}\,\overline{g} + \overline{z}\right). \tag{13}$$

Note that the update for μ_{t+1} depends on Σ_{t+1}, so these updates must be performed in the order shown above. The updates in eq. (12–13) complete the match step of BaM.

More intuition for BaM can be obtained by examining certain limiting cases of the batch size and learning rate. When $\lambda_t \to 0$, the updates have no effect, with $\Sigma_{t+1} = \Sigma_t$ and $\mu_{t+1} = \mu_t$. Alternatively, when $B = 1$ and $\lambda_t \to \infty$, the BaM updates reduce to the recently proposed updates for BBVI by (exact) Gaussian score matching (Modi et al., 2023); this equivalence is shown in Appendix C. Finally, when $B \to \infty$ and $\lambda_0 \to \infty$ (in that order), BaM converges to a Gaussian target distribution in one step; see Corollary D.5 of Appendix D.

We provide pseudocode for BaM in Algorithm 1. We note that it costs $\mathcal{O}(D^3)$ to compute the covariance update as shown in eq. (12), but for small batch sizes, when the matrix U is of rank $\mathcal{O}(B)$ with $B \ll D$, it is possible to compute the update in $\mathcal{O}(D^2B + B^3)$; this update is presented in Lemma B.3 of Appendix B.

BaM incorporates many ideas from previous work. Like the stochastic proximal point (SPP) method (Asi & Duchi, 2019; Davis & Drusvyatskiy, 2019), it minimizes a Monte Carlo estimate of a divergence subject to a regularization term. In proximal point methods, the updates are always regularized by squared Euclidean distance, but the KL divergence has been used elsewhere as a regularizer—for example, in the EM algorithm (Tseng, 2004; Chrétien & Hero, 2000) and for approximate Bayesian inference (Theis & Hoffman, 2015; Khan et al., 2015; 2016; Dai et al., 2016). KL-based regularizers are also a hallmark of mirror descent methods (Nemirovskii & Yudin, 1983), but in these methods the objective function is linearized—a poor approximation for objective functions with high curvature. Notably, BaM does not introduce any linearizations because its optimizations in eq. (8) can be solved in closed form.

3.2. Proof of convergence for Gaussian targets

In this section we analyze a concrete setting in which we can rigorously prove the convergence of the updates in Algorithm 1.

Suppose the target distribution is itself a Gaussian and the updates are computed in the limit of infinite batch size ($B \to \infty$). In this setting we show that BaM converges to the target distribution. More precisely, we show that the variational parameters converge exponentially quickly to their target values *for all fixed levels of regularization* $\lambda > 0$ *and no matter how they are initialized*. Our proof does not exclude the possibility of convergence in less restrictive settings, and in Section 5, we observe empirically that the updates also converge for non-Gaussian targets and finite batch sizes. Though the proof here does not cover such cases, it remains instructive in many ways.

To proceed, consider a Gaussian target distribution $p = \mathcal{N}(\mu_*, \Sigma_*)$. At the tth iteration of Algorithm 1, we measure the normalized *errors* in the mean and covariance parameters by

$$\varepsilon_t := \Sigma_*^{-\frac{1}{2}}(\mu_t - \mu_*), \tag{14}$$

$$\Delta_t := \Sigma_*^{-\frac{1}{2}}(\Sigma_t - \Sigma_*)\,\Sigma_*^{-\frac{1}{2}}. \tag{15}$$

The theorem below shows that $\varepsilon_t, \Delta_t \to 0$ in spectral norm. Specifically, it shows that this convergence occurs exponentially quickly at a rate controlled by the quality of initialization and amount of regularization.

Theorem 3.1 (Exponential convergence). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$ in Algorithm 1, and let $\alpha > 0$ denote the minimum eigenvalue of the matrix $\Sigma_*^{-\frac{1}{2}}\Sigma_0\Sigma_*^{-\frac{1}{2}}$. For any fixed level of regularization $\lambda > 0$, define

$$\beta := \min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right), \quad \delta := \frac{\lambda\beta}{1+\lambda}, \tag{16}$$

where $\beta \in (0, 1]$ measures the quality of initialization and $\delta \in (0, 1)$ denotes a rate of decay. Then with probability 1 in the limit of infinite batch size ($B \to \infty$), and for all $t \geq 0$, the normalized errors in eqs. (14–15) satisfy

$$\|\varepsilon_t\| \leq (1-\delta)^t \|\varepsilon_0\|, \tag{17}$$

$$\|\Delta_t\| \leq (1-\delta)^t \|\Delta_0\| \;+\; t(1-\delta)^{t-1}\|\varepsilon_0\|^2. \tag{18}$$

Before sketching the proof we make three remarks. First, these error bounds behave sensibly: they suggest that the updates converge more slowly when the learning rate is small (with $\lambda \ll 1$), when the variational mean is poorly initialized (with $\|\varepsilon_0\|^2 \gg 1$), and/or when the initial estimate of the covariance is nearly singular (with $\alpha \ll 1$). Second, the theorem holds under very general conditions—not only for any initialization of μ_0 and $\Sigma_0 \succ 0$, but also for any $\lambda > 0$. This robustness is typical of *proximal* algorithms, which are well-known for their stability with respect to hyperparameters (Asi & Duchi, 2019), but it is uncharacteristic of many *gradient-based* methods, which only converge when the learning rate varies inversely with the largest eigenvalue of an underlying Hessian (Garrigos & Gower, 2023). Third, with more elaborate bookkeeping, we can derive *tighter* bounds both for the above setting and also when different iterations use varying levels of regularization $\{\lambda_t\}_{t=0}^{\infty}$. We give a full proof with these extensions in Appendix D.

Proof Sketch. The crux of the proof is to bound the normalized errors in eqs. (14–15) from one iteration to the next. Most importantly, we show that

$$\|\varepsilon_{t+1}\| \leq (1-\delta)\|\varepsilon_t\|, \tag{19}$$

$$\|\Delta_{t+1}\| \leq (1-\delta)\|\Delta_t\| \;+\; \|\varepsilon_t\|^2, \tag{20}$$

where δ is given by eq. (16), and from these bounds, we use induction to prove the overall rates of decay in eqs. (17-18). Here we briefly describe the steps that are needed to derive the bounds in eqs. (19–20).

The first is to examine the statistics computed at each iteration of the algorithm in the *infinite batch* limit ($B \to \infty$). This limit is simplifying because by the law of large numbers, we can replace the batched averages over B samples at each iteration by their expected values under the variational distribution $q_t = \mathcal{N}(\mu_t, \Sigma_t)$. The second step of the proof is to analyze the algorithm's convergence in terms

of the *normalized* mean ε_t in eq. (14) and the *normalized* covariance matrix

$$J_t = \Sigma_*^{-\frac{1}{2}} \Sigma_t \Sigma_*^{-\frac{1}{2}} = I + \Delta_t, \tag{21}$$

where I denotes the identity matrix. In the infinite batch limit, we show that with probability 1 these quantities satisfy

$$\lambda J_{t+1} \left(J_t + \tfrac{1}{1+\lambda} \varepsilon_t \varepsilon_t^\top \right) J_{t+1} + J_{t+1} = (1+\lambda) J_t, \tag{22}$$

$$\varepsilon_{t+1} = \left(I - \tfrac{\lambda}{1+\lambda} J_{t+1} \right) \varepsilon_t. \tag{23}$$

The third step of the proof is to *sandwich* the matrix J_{t+1} that appears in eq. (22) between two other positive-definite matrices whose eigenvalues are more easily bounded. Specifically, at each iteration t, we introduce matrices H_{t+1} and K_{t+1} defined by

$$\lambda H_{t+1} \left(J_t + \tfrac{\|\varepsilon_t\|^2}{1+\lambda} I \right) H_{t+1} + H_{t+1} = (1+\lambda) J_t, \tag{24}$$

$$\lambda K_{t+1} \, J_t \, K_{t+1} + K_{t+1} = (1+\lambda) J_t. \tag{25}$$

It is easier to analyze the solutions to these equations because they replace the outer-product $\varepsilon_t \varepsilon_t^\top$ in eq. (22) by a multiple of the identity matrix. We show that for all times $t \geq 0$,

$$H_{t+1} \preceq J_{t+1} \preceq K_{t+1}, \tag{26}$$

so that we can prove $\|J_t - I\| \to 0$ by showing $\|H_t - I\| \to 0$ and $\|K_t - I\| \to 0$. Finally, the last (and most technical) step is to derive the bounds in eqs. (19–20) by combining the sandwich inequality in eq. (26) with a detailed analysis of eqs. (22–25). □

4. Related work

BaM builds on intuitions from earlier work on Gaussian score matching (GSM) (Modi et al., 2023). GSM is an iterative algorithm for BBVI that updates a full-covariance Gaussian by analytically solving a system of nonlinear equations. As previously discussed, BaM recovers GSM as a special limiting case. A limitation of GSM is that it aims to match the scores exactly; thus, if the target is not exactly Gaussian, the updates for GSM attempt to solve an infeasible problem, In addition, the batch updates for GSM perform an ad hoc averaging that is not guaranteed to match any scores exactly, even when it is possible to do so. BaM overcomes these limitations by optimizing a proper score-based divergence on each batch of samples. Empirically, with BaM, we observe that larger batch sizes lead to more stable convergence. The score-based divergence behind BaM also lends itself to analysis, and we can provide theoretical guarantees on the convergence of BaM for Gaussian targets.

Proximal point methods have been studied in several papers in the context of variational inference; typically the objective is a stochastic estimate of the ELBO with a (forward) KL regularization term. For example, Theis & Hoffman (2015) optimize this objective using alternating coordinate ascent. In other work, Khan et al. (2015; 2016) propose a splitting method for this objective, and by linearizing the difficult terms, they obtain a closed-form solution when the variational family is Gaussian and additional knowledge is given about the structure of the target. By contrast, BaM does not resort to linearization in order to obtain an analytical solution, nor does it require additional assumptions on the structure of the target.

Proximal algorithms have also been developed for Gaussian variational families based on the Wasserstein metric. Lambert et al. (2022) consider a KL objective with the Wasserstein metric as a regularizer; in this case, the proximal step is not solvable in closed form. On the other hand, Diao et al. (2023) consider a proximal-gradient method, and show that the proximal step admits a closed-form solution.

Several works consider score matching with a Fisher divergence in the context of VI. For instance, Yu & Zhang (2023) propose a score-matching approach for semi-implicit variational families based on stochastic gradient optimization of the Fisher divergence. Zhang et al. (2018) use the Fisher divergence with an energy-based model as the variational family. BaM differs from these approaches by working with a Gaussian variational family and an affine-invariant score-based divergence.

Finally, we note that the idea of score matching (Hyvärinen, 2005) with a (weighted) Fisher divergence appears in many contexts beyond VI (Song & Ermon, 2019; Barp et al., 2019). One such context is generative modeling: here, given a set of training examples, the goal is to approximate an unknown data distribution p by a parameterized model p_θ with an intractable normalization constant. Note that in this setting one can evaluate $\nabla \log p_\theta$ but not $\nabla \log p$. This setting is quite different from the setting of VI in this paper where we do *not* have samples from p, where we *can* evaluate $\nabla \log p$, and where the approximating distribution q has the much simpler and more tractable form of a multivariate Gaussian.

5. Experiments

We evaluate BaM against two other BBVI methods for Gaussian variational families with full covariance matrices. The first of these is automatic differentiation VI (ADVI) (Kucukelbir et al., 2017), which is based on ELBO maximization, and the second is GSM (Modi et al., 2023), as described in the previous section. We implement all algorithms using `JAX` (Bradbury et al., 2018),[1] which supports efficient automatic differentiation both on CPU and GPU. We provide pseudocode for these methods in Appendix E.1.

[1] Python implementations of BaM and the baselines are available at: https://github.com/modichirag/GSM-VI/.

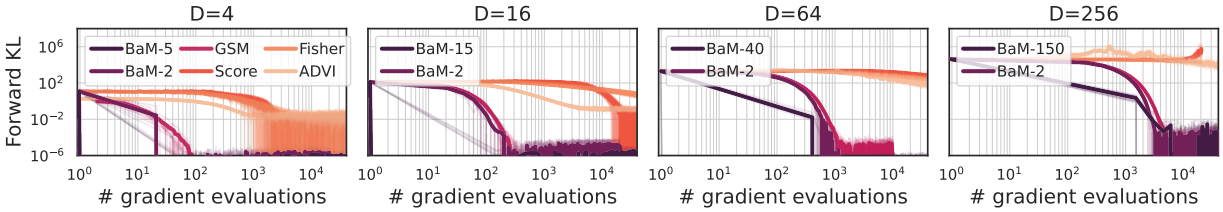


Figure 5.1: Gaussian targets of increasing dimension. Solid curves indicate the mean over 10 runs (transparent curves). ADVI, Score, Fisher, and GSM use a batch size of $B=2$. The batch size for BaM is given in the legend.

5.1. Synthetically-constructed target distributions

We first validate BaM in two settings where we know the true target distribution p. In the first setting, we construct Gaussian targets with increasing number of dimensions. In the second setting, we study BaM for distributions with increasing (but controlled) amounts of non-Gaussianity. As evaluation metrics, we use empirical estimates of the KL divergence in both the forward direction, $\mathrm{KL}(p;q)$, and the reverse direction, $\mathrm{KL}(q;p)$.

Gaussian targets with increasing dimensions. We construct Gaussian targets of increasing dimension with $D = 4, 16, 64, 256$. In Figure 5.1, we compare BaM, ADVI, and GSM on each of these target distributions, plotting the forward KL divergence against the number of gradient evaluations; here we also consider two modified ADVI methods, where instead of the ELBO loss, we use the score-based divergence (labeled as "Score") and the Fisher divergence (labeled as "Fisher"). Results for the reverse KL divergence and other parameter settings are provided in Appendix E.3. In all of these experiments, we use a constant learning rate $\lambda_t = BD$ for BaM. Overall, we find that BaM converges orders of magnitude faster than ADVI. While GSM is competitive with BAM in some experiments, BaM converges more quickly with increasing batch size; this is unlike GSM which was observed to have marginal gains beyond $B=2$ for Gaussian targets (Modi et al., 2023).

We also observe that the gradient-based methods (ADVI, Score, Fisher) have similar performance in terms of convergence, and the score-based divergence is typically more sensitive to the learning rate. In Appendix E.2, we present wallclock timings for the methods, which show that the gradient evaluations dominate the computational cost in lower-dimensional settings.

Non-Gaussian targets with varying skew and tails. The sinh-arcsinh normal distribution transforms a Gaussian random variable via the hyperbolic sine function and its inverse (Jones & Pewsey, 2009; 2019). If $y \sim \mathcal{N}(\mu, \Sigma)$, then a sample from the sinh-arcsinh normal distribution is

$$z = \sinh\left(\frac{1}{\tau}\left(\sinh^{-1}(y) + s\right)\right),$$

where the parameters $s \in \mathbb{R}$ and $\tau > 0$ control, respectively, the skew and the heaviness of the tails. The Gaussian distribution is recovered when $s=0$ and $\tau=1$.

We construct different non-Gaussian target distributions by varying these parameters. The results are presented in Figure 5.2 and Figure E.4. Here we use a decaying learning rate $\lambda_t = BD/(t+1)$ for BaM, as some decay is necessary for BaM to converge when the target distribution is non-Gaussian.

First, we construct target distributions with normal tails ($t = 1$) but varying skew ($s = 0.2, 1.0, 1.8$). Here we observe that BaM converges faster than ADVI. For large skew ($s = 1.0, 1.8$), BaM converges to a higher value of the forward KL divergence but to similar values of the reverse KL divergence. In these experiments, we see that GSM and ADVI often have similar performance but that BaM stabilizes more quickly with larger batch sizes. Notably, the reverse KL divergence for GSM diverges when the target distribution is highly skewed ($s=1.8$). The Score method diverges for highly skewed targets as well, and we found this method to be more sensitive to the learning rate.

Next we construct target distributions with no skew ($s=0$) but tails of varying heaviness ($t = 0.1, 0.9, 1.7$). Here we find that all methods tend to converge to similar values of the reverse KL divergence. In some cases, BaM and ADVI converge to better values than GSM, and BaM typically converges in fewer gradient evaluations than ADVI.

5.2. Application: hierarchical Bayesian models

We now consider the application of BaM to posterior inference. Suppose we have observations $\{x_n\}_{n=1}^N$, and the target distribution is the posterior density

$$p\left(z \mid \{x_n\}_{n=1}^N\right) \propto p(z)\, p\left(\{x_n\}_{n=1}^N \mid z\right), \qquad (27)$$

with prior $p(z)$ and likelihood $p(\{x_n\}_{n=1}^N \mid z)$. We examine three target distributions from `posteriordb` (Magnusson et al., 2022), a database of `Stan` (Carpenter et al., 2017; Roualdes et al., 2023) models with reference samples generated using Hamiltonian Monte Carlo (HMC). The first target is nearly Gaussian (`arK`, $D = 7$). The other

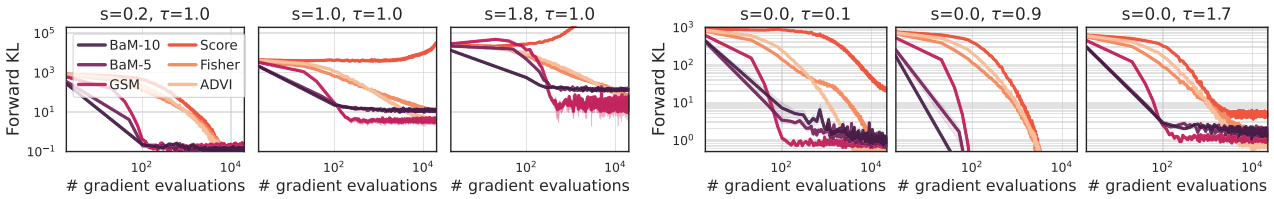


Figure 5.2: Non-Gaussian targets constructed using the sinh-arcsinh distribution, varying the skew s and the tail weight t. The curves denote the mean of the forward KL divergence over 10 runs, and shaded regions denote their standard error. ADVI, Score, Fisher, and GSM use a batch size of $B=5$.

two targets are non-Gaussian: one is a Gaussian process (GP) Poisson regression model (`gp-pois-regr`, $D=13$), and the other is the 8-schools hierarchical Bayesian model (`eight-schools-centered`, $D=10$).

In these experiments, we evaluate BaM, ADVI, and GSM by computing the relative errors of the posterior mean and standard deviation (SD) estimates with respect to those from HMC samples (Welandawe et al., 2022); we define these quantities and present additional results in Appendix E.5. We use a decaying learning rate $\lambda_t = BD/(t+1)$ for BaM.

Figure 5.3 compares the relative mean errors of BaM, ADVI, and GSM for batch sizes $B=8$ and $B=32$. We observe that BaM outperforms ADVI. For smaller batch sizes GSM can converge faster than BaM, but it oscillates around the solution. BaM performs better with increasing batch size, converging more quickly and to a more stable result, while GSM and ADVI do not benefit from increasing batch size. In the appendix, we report the relative SD error and find similar results except that in the hierarchical example, BaM converges to a larger relative SD error.

5.3. Application: deep generative model

In a deep generative model, the likelihood is parameterized by the output of a neural network Ω, e.g.,

$$z_n \sim \mathcal{N}(0, I) \tag{28}$$

$$x_n \mid z_n \sim \mathcal{N}(\Omega(z_n, \hat{\theta}), \sigma^2 I), \tag{29}$$

where x_n corresponds to a high-dimensional object, such as an image, and z_n is a low-dimensional representation of x_n. The neural network Ω is parameterized by $\hat{\theta}$ and maps z_n to the mean of the likelihood $p(x_n|z_n)$. For this example, we set $\sigma^2 = 0.1$. The above joint distribution underlies many deep learning models (Tomczak, 2022), including the variational autoencoder (Kingma & Welling, 2014; Rezende et al., 2014). We train the neural network on the CIFAR-10 image data set (Krizhevsky, 2009). We model the images as continuous, with $x_n \in \mathbb{R}^{3072}$, and learn a latent representation $z_n \in \mathbb{R}^{256}$; see Appendix E.6 for details.

Given a new observation x', we wish to approximate the posterior $p(z'|x')$. As an evaluation metric, we examine how well x' is reconstructed by feeding the posterior expectation $\mathbb{E}[z'|x']$ into the neural network $\Omega(\cdot, \hat{\theta})$. The quality of the reconstruction is assessed visually and using the mean squared error (MSE, Figure 5.4); we present the MSE plotted against wallclock time in Figure E.7. For ADVI and BaM, we use a pilot run of $T=100$ iterations to find a suitable learning rate; we then run the algorithms for $T=1000$ iterations. (GSM does not require this tuning step.) BaM performs poorly when the batch size is very small ($B=10$) relative to the dimension of the latent variable z', but it becomes competitive as the batch size is increased. When the batch size is comparable to the dimension of z_n (i.e. $B=300$), BaM converges an order of magnitude (or more) faster than ADVI and GSM.

To refine our comparison, suppose we have a computational budget of 3000 gradient evaluations. Under this budget, ADVI achieves its lowest MSE for $B=10$ and $T=300$, while BaM produces a comparable result for $B=300$ and $T=10$. Hence, the gradient evaluations for BaM can be largely parallelized. By contrast, most gradients for ADVI must be evaluated sequentially. Notably, Figure E.7 shows that BaM with $B=300$ converges faster in wallclock time.

Depending on how the parameter $\hat{\theta}$ of the neural network is estimated, it possible to learn an encoder and perform amortized variational inference (AVI) on a new observation x'. When such an encoder is available, estimations of $p(z'|x')$ can be obtained essentially for free. In our experiment, both BaM and ADVI eventually achieve a lower reconstruction error than AVI. This result is expected because our AVI implementation uses a factorized Gaussian approximation, whereas BaM and ADVI use a full-covariance approximation, and the latter provides better compression of x' even though the dimension of z' and the weights of the neural network remain unchanged.

6. Discussion and future work

In this paper, we introduce a score-based divergence that is especially well-suited to BBVI with Gaussian variational families. We show that the score-based divergence has a number of desirable properties. We then propose a regular-

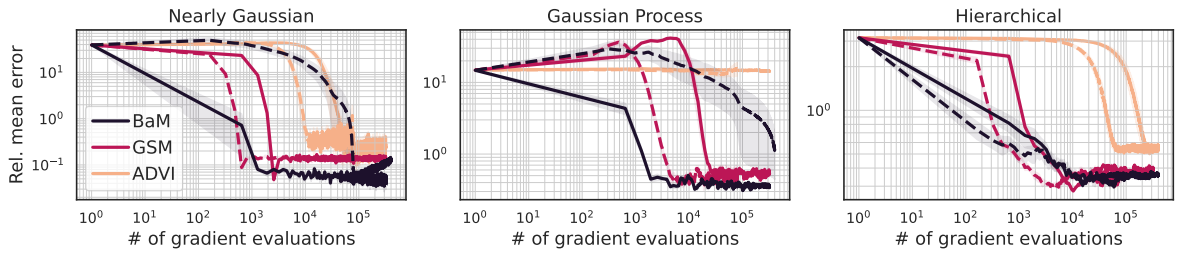
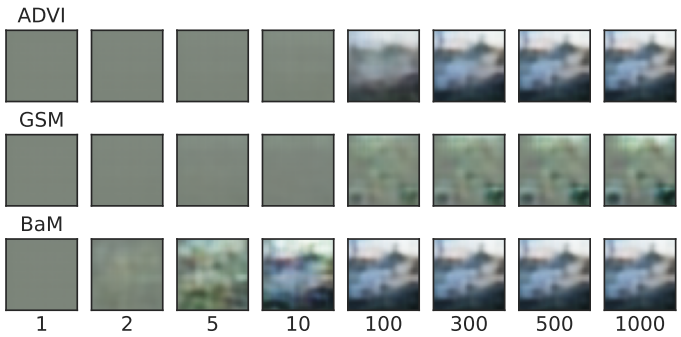
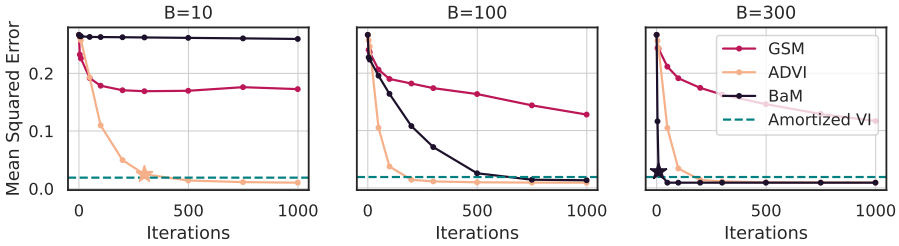


Figure 5.3: Posterior inference in Bayesian models. The curves denote the mean over 5 runs, and shaded regions denote their standard error. Solid curves ($B=32$) correspond to larger batch sizes than dashed curves ($B=8$).



(a) Image reconstruction across iterations with batch size, $B = 300$.



(b) Reconstruction error for varying batch sizes.

Figure 5.4: Image reconstruction and error when the posterior mean of z' is fed into the generative neural network. The beige and purple stars highlight the best outcome for ADVI and BaM, respectively, after 3,000 gradient evaluations.

ized optimization based on this divergence, and we show that it admits a closed-form solution, leading to a fast iterative algorithm for score-based BBVI. We analyze the convergence of score-based BBVI when the target is Gaussian, and in the limit of an infinite batch size, we show that the updates converge exponentially quickly to the target mean and covariance. Finally, we demonstrate the effectiveness of BaM in a number of empirical studies involving both Gaussian and non-Gaussian targets; here we observe that for sufficiently large batch sizes, our method converges much faster than other BBVI algorithms.

There are a number of fruitful directions for future work. First, it remains to analyze the convergence of BaM in the finite-batch case and for a larger class of target distributions. Second, it seems promising to develop score-based BBVI for other (non-Gaussian) variational families, and more generally, to study what divergences lend themselves to stochastic proximal point algorithms. Third, the BaM approach can be modified to utilize data subsampling (potentially with control variates (Wang et al., 2024)) for large-scale Bayesian inference problems, where a noisy estimate of the target density's score is used in place of its exact score.

Finally, we note that the score-based divergence, which is computable for unnormalized models, has useful applications beyond VI (Hyvärinen, 2005); e.g., the affine invariance property makes it attractive as a goodness-of-fit diagnostic for inference methods. Further study remains to characterize the relationship of the score-based divergence to other such diagnostics (Gorham & Mackey, 2015; Liu et al., 2016; Barp et al., 2019; Welandawe et al., 2022).

Acknowledgements

We thank Bob Carpenter, Ryan Giordano, and Yuling Yao for helpful discussions and anonymous reviewers for their feedback on the paper. This work was supported in part by NSF IIS-2127869, NSF DMS-2311108, NSF/DoD PHY-2229929, ONR N00014-17-1-2131, ONR N00014-15-1-2209, the Simons Foundation, and Open Philanthropy.

Impact statement

This paper presents work whose goal is to advance approximate probabilistic inference. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Asi, H. and Duchi, J. C. Stochastic (approximate) proximal point methods: convergence, optimality, and adaptivity. *SIAM Journal on Optimization*, 29(3):2257–2290, 2019. ISSN 1052-6234.

Barp, A., Briol, F.-X., Duncan, A., Girolami, M., and Mackey, L. Minimum Stein discrepancy estimators. *Advances in Neural Information Processing Systems*, 32, 2019.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

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

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Burroni, J., Domke, J., and Sheldon, D. Sample average approximation for black-box VI. *arXiv preprint arXiv:2304.06803*, 2023.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017.

Chrétien, S. and Hero, A. O. Kullback proximal algorithms for maximum-likelihood estimation. *IEEE Transactions on Information Theory*, 46(5):1800–1810, 2000.

Dai, B., He, N., Dai, H., and Song, L. Provable Bayesian inference via particle mirror descent. In *Artificial Intelligence and Statistics*, pp. 985–994. PMLR, 2016.

Davis, D. and Drusvyatskiy, D. Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.*, 29(1):207–239, 2019.

Dhaka, A. K., Catalina, A., Andersen, M. R., Magnusson, M., Huggins, J., and Vehtari, A. Robust, accurate stochastic optimization for variational inference. *Advances in Neural Information Processing Systems*, 33, 2020.

Dhaka, A. K., Catalina, A., Welandawe, M., Andersen, M. R., Huggins, J., and Vehtari, A. Challenges and opportunities in high dimensional variational inference. *Advances in Neural Information Processing Systems*, 34, 2021.

Diao, M. Z., Balasubramanian, K., Chewi, S., and Salim, A. Forward-backward Gaussian variational inference via JKO in the Bures-Wasserstein space. In *International Conference on Machine Learning*. PMLR, 2023.

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

Domke, J., Garrigos, G., and Gower, R. Provable convergence guarantees for black-box variational inference. *Advances in Neural Information Processing Systems*, 36, 2023.

Garrigos, G. and Gower, R. M. Handbook of convergence theorems for (stochastic) gradient methods, 2023.

Giordano, R., Ingram, M., and Broderick, T. Black box variational inference with a deterministic objective: Faster, more accurate, and even more black box. *Journal of Machine Learning Research*, 25(18):1–39, 2024.

Gorham, J. and Mackey, L. Measuring sample quality with Stein's method. *Advances in Neural Information Processing Systems*, 28, 2015.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Jones, C. and Pewsey, A. Sinh-arcsinh distributions. *Biometrika*, 96(4):761–780, 2009.

Jones, C. and Pewsey, A. The sinh-arcsinh normal distribution. *Significance*, 16(2):6–7, 2019.

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

Khan, M. E., Baqué, P., Fleuret, F., and Fua, P. Kullback-Leibler proximal variational inference. In *Advances in Neural Information Processing Systems*, 2015.

Khan, M. E., Babanezhad, R., Lin, W., Schmidt, M., and Sugiyama, M. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

Kim, K., Oh, J., Wu, K., Ma, Y., and Gardner, J. R. On the convergence of black-box variational inference. *Advances in Neural Information Processing Systems*, 36, 2023.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 2017.

Kučera, V. On nonnegative definite solutions to matrix quadratic equations. *Automatica*, 8(4):413–423, 1972a.

Kučera, V. A contribution to matrix quadratic equations. *IEEE Transactions on Automatic Control*, 17(3):344–347, 1972b.

Lambert, M., Chewi, S., Bach, F., Bonnabel, S., and Rigollet, P. Variational inference via Wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 35, 2022.

Liu, Q., Lee, J., and Jordan, M. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*. PMLR, 2016.

Locatello, F., Dresdner, G., Khanna, R., Valera, I., and Rätsch, G. Boosting black box variational inference. *Advances in Neural Information Processing Systems*, 31, 2018.

Magnusson, M., Bürkner, P., and Vehtari, A. posteriordb: a set of posteriors for Bayesian inference and probabilistic programming. https://github.com/stan-dev/posteriordb, November 2022.

Modi, C., Margossian, C., Yao, Y., Gower, R., Blei, D., and Saul, L. Variational inference with Gaussian score matching. *Advances in Neural Information Processing Systems*, 36, 2023.

Nemirovskii, A. and Yudin, D. B. *Problem complexity and method efficiency in optimization*. John Wiley and Sons, 1983.

Potter, J. E. Matrix quadratic solutions. *SIAM Journal of Applied Mathematics*, 14(3):496–501, 1966.

Qiao, Y. and Minematsu, N. A study on invariance of f-divergence and its application to speech recognition. *IEEE Transactions on Signal Processing*, 58(7):3884–3890, 2010.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822. PMLR, 2014.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*. PMLR, 2014.

Roualdes, E., Ward, B., Axen, S., and Carpenter, B. BridgeStan: Efficient in-memory access to Stan programs through Python, Julia, and R. https://github.com/roualdes/bridgestan, March 2023.

Ryder, T., Golightly, A., McGough, A. S., and Prangle, D. Black-box variational inference for stochastic differential equations. In *International Conference on Machine Learning*. PMLR, 2018.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

Shurbet, G., Lewis, T., and Boullion, T. Quadratic matrix equations. *The Ohio Journal of Science*, 74(5), 1974.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Theis, L. and Hoffman, M. A trust-region method for stochastic variational inference with applications to streaming data. In *International Conference on Machine Learning*. PMLR, 2015.

Tomczak, J. M. *Deep Generative Modeling*. Springer, 2022.

Tseng, P. An analysis of the EM algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29(1):27–44, 2004.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Wang, X., Geffner, T., and Domke, J. Dual control variate for faster black-box variational inference. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Welandawe, M., Andersen, M. R., Vehtari, A., and Huggins, J. H. A framework for improving the reliability of black-box variational inference. *arXiv preprint arXiv:2203.15945*, 2022.

Yu, L. and Zhang, C. Semi-implicit variational inference via score matching. In *International Conference on Learning Representations*, 2023.

Yuan, Y., Liu, L., Zhang, H., and Liu, H. The solutions to the quadratic matrix equation X*AX+B*X+D=0. *Applied Mathematics and Computation*, 410:126463, 2021.

Zhang, C., Shahbaba, B., and Zhao, H. Variational Hamiltonian Monte Carlo via score matching. *Bayesian Analysis*, 13(2):485, 2018.

A. Score-based divergence

In Section 2 we introduced a score-based divergence between two distributions, p and q, over $\mathbb{R}^D$, and specifically we considered the case where q was Gaussian. In this section, we define this score-based divergence more generally. In particular, here we assume only that these distributions satisfy the following properties:

(i) $p(z) > 0$ and $q(z) > 0$ for all $z \in \mathbb{R}^D$.
(ii) ∇p and ∇q exist and are continuous everywhere in $\mathbb{R}^D$.
(iii) $\mathbb{E}_q \left[\|\nabla \log q\|^2 \right] < \infty$.

There may be weaker properties than these that also yield the following results (or various generalizations thereof), but the above will suffice for our purposes.

This appendix is organized as follows. We begin with a lemma that is needed to define a score-based divergence for distributions (not necessarily Gaussian) satisfying the above properties. We then show that this score-based divergence has several appealing properties in its own right: it is nonnegative and invariant under affine reparameterizations, it takes a simple and intuitive form for distributions that are related by annealing or exponential tilting, and it reduces to the KL divergence in certain special cases.

Lemma A.1. The matrix defined by $\Gamma_q = \mathbb{E}_q \left[(\nabla \log q)(\nabla \log q)^\top \right]$ exists in $\mathbb{R}^{D \times D}$ and is positive definite.

Proof. Let u be any unit vector in $\mathbb{R}^D$. We shall prove the theorem by showing that $0 < u^\top \Gamma_q u < \infty$, or equivalently that all of the eigenvalues of Γ_q are finite and positive. The boundedness follows easily from property (iii) since

$$u^\top \Gamma_q u \;=\; \mathbb{E}_q \left[(\nabla \log q \cdot u)^2 \right] \;\leq\; \mathbb{E}_q \left[\|\nabla \log q\|^2 \right] \;< \infty. \tag{30}$$

To show positivity, we appeal to property (ii) that q is differentiable; hence for all $t > 0$ we can write

$$q(tu) = q(0) + \int_0^t d\tau \; u^\top \nabla q(\tau u) = q(0) + \int_0^t d\tau \; q(\tau u) \, \nabla \log q(\tau u) \cdot u. \tag{31}$$

To proceed, we take the limit $t \to \infty$ on both sides of this equation, and we appeal to property (i) that $q(0) > 0$. Moreover, since $\lim_{t \to \infty} q(tu) = 0$ for all normalizable distributions q, we see that

$$\int_0^\infty d\tau \; q(\tau u) \, \nabla \log q(\tau u) \cdot u < 0. \tag{32}$$

For this inequality to be satisfied, there must exist some $t_0 \geq 0$ such that $\nabla \log q(t_0 u) \cdot u < 0$. Let $z_0 = t_0 u$, and let $\delta = -\nabla \log q(z_0) \cdot u$. Since q and ∇q are continuous by properties (iii-iv), there must exist some finite ball $\mathcal{B}$ around z_0 such that $\nabla \log q(z) \cdot u < -\frac{\delta}{2}$ for all $z \in \mathcal{B}$. Let $q_{\mathcal{B}} = \min_{z \in \mathcal{B}} q(z)$, and note that $q_{\mathcal{B}} > 0$ since it is the minimum of a positive-valued function on a compact set. It follows that

$$u^\top \Gamma_q u = \mathbb{E}_q \left[(\nabla \log q \cdot u)^2 \right] > q_{\mathcal{B}} \cdot \mathrm{vol}(\mathcal{B}) \cdot \left(\tfrac{\delta}{2} \right)^2 > 0, \tag{33}$$

where the inequality is obtained by considering only those contributions to the expected value from within the volume of the ball $\mathcal{B}$ around z_0. This proves the lemma. □

The lemma is needed for the following definition of the score-based divergence. Notably, the definition assumes that the matrix $\mathbb{E}_q \left[(\nabla \log q)(\nabla \log q)^\top \right]$ is invertible.

Definition A.2 (Score-based divergence). Let p and q satisfy the properties listed above, and let Γ_q be defined as in Lemma A.1. Then we define the *score-based divergence* between q and p as

$$\mathscr{D}(q; p) = \mathbb{E}_q \left[\left(\nabla \log \tfrac{q}{p} \right)^\top \Gamma_q^{-1} \left(\nabla \log \tfrac{q}{p} \right) \right]. \tag{34}$$

Let us quickly verify that this definition reduces to the previous one in Section 2 where q is assumed to be Gaussian. In particular, suppose that $q = \mathcal{N}(\nu, \Psi)$. In this case

$$\Gamma_q = \mathbb{E}_q[(\nabla \log q)(\nabla \log q)^\top] = \mathbb{E}_q[\Psi^{-1}(z-\nu)(z-\nu)^\top \Psi^{-1}] = \Psi^{-1}\Psi\Psi^{-1} = \Psi^{-1} = [\mathrm{Cov}(q)]^{-1}. \tag{35}$$

Substituting this result into eq. (34), we recover the more specialized definition of the score-based divergence in Section 2.

We now return to the more general definition in eq. (34). Next we show this score-based divergence shares many desirable properties with the Kullback-Leibler divergence; indeed, in certain special cases of interest, these two divergences, $\mathscr{D}(q;p)$ and $\mathrm{KL}(q;p)$, are equivalent. These properties are demonstrated in the following theorems.

Theorem A.3 (Nonnegativity). $\mathscr{D}(q;p) \geq 0$ with equality if and only if $p(z) = q(z)$ for all $z \in \mathbb{R}^D$.

Proof. Nonnegativity follows from the previous lemma, and it is clear that the divergence vanishes if $p = q$. To prove the converse, we note that for any $z \in \mathbb{R}^D$, we can write

$$\log \frac{p(z)}{q(z)} \;=\; \log \frac{p(0)}{q(0)} + \int_0^1 dt\, \nabla \log \left[\frac{p(tz)}{q(tz)}\right] \cdot z. \tag{36}$$

Now suppose that $\mathscr{D}(q;p) = 0$. Then it must be the case that $\nabla \log p = \nabla \log q$ everywhere in $\mathbb{R}^D$. (If it were the case that $\nabla \log p(z_0) \neq \nabla \log q(z_0)$ for some $z_0 \in \mathbb{R}^D$, then by continuity, there would also exist some ball around z_0 where these gradients were not equal; furthermore, in this case, the value inside the expectation of eq. (34) would be positive everywhere inside this ball, yielding a positive value for the divergence.) Since the gradients of $\log p$ and $\log q$ are everywhere equal, it follows from eq. (36) that

$$\log \frac{p(z)}{q(z)} \;=\; \log \frac{p(0)}{q(0)}, \tag{37}$$

or equivalently, that $p(z)$ and $q(z)$ have some constant ratio independent of z. But this constant ratio must be equal to one because both distributions yield the same value when they are integrated over $\mathbb{R}^D$. □

Theorem A.4 (Affine invariance). Let $f : \mathbb{R}^D \to \mathbb{R}^D$ be an affine transformation, and consider the induced densities $\tilde{q}(f(z)) = q(z)|\mathcal{J}(z)|^{-1}$ and $\tilde{p}(f(z)) = p(z)|\mathcal{J}(z)|^{-1}$, where $\mathcal{J}(z)$ is the determinant of the Jacobian of f. Then $\mathscr{D}(q;p) = \mathscr{D}(\tilde{q};\tilde{p})$.

Proof. Denote the affine transformation by $\tilde{z} = Az + b$ where $A \in \mathbb{R}^{D\times D}$ and $b \in \mathbb{R}^D$. Then we have

$$\nabla_z\big[\log p(z)\big] = \nabla_z\left[\log\left(\tilde{p}(\tilde{z})\left|\frac{d\tilde{z}}{dz}\right|\right)\right] = \nabla_z[\log\left(\tilde{p}(\tilde{z})\,|A|\right)] = \left(\frac{d\tilde{z}}{dz}\right)^\top \nabla_{\tilde{z}}[\log \tilde{p}(\tilde{z})] = A^\top \nabla_{\tilde{z}}\big[\log \tilde{p}(\tilde{z})\big], \tag{38}$$

and a similar relation holds for $\nabla_x \log q(z)$. It follows that

$$\begin{aligned}
\mathscr{D}(q;p) &= \mathbb{E}_q\Big[(\nabla \log p - \nabla \log q)^\top \left(\mathbb{E}_q\big[(\nabla \log q)(\nabla \log q)^\top\big]\right)^{-1} (\nabla \log p - \nabla \log q)\Big] && (39)\\
&= \mathbb{E}_{\tilde{q}}\Big[(\nabla \log \tilde{p} - \nabla \log \tilde{q})^\top A\big(A^\top \mathbb{E}_{\tilde{q}}\big[(\nabla \log \tilde{q})(\nabla \log \tilde{q})^\top\big]A\big)^{-1} A^\top (\nabla \log \tilde{p} - \nabla \log \tilde{q})\Big] && (40)\\
&= \mathbb{E}_{\tilde{q}}\Big[(\nabla \log \tilde{p} - \nabla \log \tilde{q})^\top \left(\mathbb{E}_{\tilde{q}}\big[(\nabla \log \tilde{q})(\nabla \log \tilde{q})^\top\big]\right)^{-1} (\nabla \log \tilde{p} - \nabla \log \tilde{q})\Big] && (41)\\
&= \mathscr{D}\big(\tilde{q}, \tilde{p}\big). && (42)
\end{aligned}$$

Note the important role played by the matrix $\Gamma_q = \mathbb{E}_q\big[(\nabla \log q)(\nabla \log q)^\top\big]$ in this calculation. In particular, the unscaled quantity $\mathbb{E}_q[\|\nabla \log p - \nabla \log q\|^2]$ is not invariant under affine reparameterizations of $\mathbb{R}^D$. □

Theorem A.5 (Annealing). If p is an annealing of q, with $p \propto q^\beta$, then $\mathscr{D}(q;p) = D(\beta-1)^2$.

Proof. In this case $\nabla \log p = \beta \nabla \log q$. Thus, with Γ_q defined as in Lemma A.1, we have

$$\mathscr{D}(q;p) = (\beta-1)^2\, \mathbb{E}_q \left[(\nabla \log q)^\top \Gamma_q^{-1} (\nabla \log q) \right] = (\beta-1)^2 \operatorname{tr}\left(\Gamma_q^{-1}\Gamma_q\right) = D(\beta-1)^2. \tag{43}$$

Here we see that $\mathscr{D}(q;p)$ measures the *difference in inverse temperature* from the annealing. Note that in the limit $\beta \to 0$ of a uniform distribution, eq. (43) yields a divergence of D that is independent of the base distribution q. □

Theorem A.6 (Exponential tilting). If p is an exponential tilting of q, with $p(z) \propto q(z)\, e^{\theta^\top z}$, then $\mathscr{D}(q;p) = \theta^\top \Gamma_q^{-1} \theta$ where Γ_q is defined as in Lemma A.1.

Proof. In this case $\nabla \log p - \nabla \log q = \theta$, and the result follows at once from substitution into eq. (34). □

Proposition A.7 (Gaussian score-based divergences). Suppose that p is multivariate Gaussian with mean μ and covariance Σ and that q is multivariate Gaussian with mean ν and covariance Ψ, respectively. Then

$$\mathscr{D}(q;p) = \operatorname{tr}\left[\left(I - \Psi\Sigma^{-1}\right)^2 \right] + (\nu-\mu)^\top \Sigma^{-1} \Psi \Sigma^{-1} (\nu-\mu). \tag{44}$$

Proof. We use the previous result in eq. (35) that $\Gamma_q = \Psi^{-1}$ when q is Gaussian with covariance Ψ. Then from eq. (34) the score-based divergence is given by

$$\begin{aligned}
\mathscr{D}(q;p) &= \mathbb{E}_q \left[(\nabla \log p - \nabla \log q)^\top \Gamma_q^{-1} (\nabla \log p - \nabla \log q) \right], && (45)\\
&= \mathbb{E}_q \left[\left(\Sigma^{-1}(z-\mu) - \Psi^{-1}(z-\nu)\right)^\top \Psi \left(\Sigma^{-1}(z-\mu) - \Psi^{-1}(z-\nu)\right) \right], && (46)\\
&= \mathbb{E}_q \left[\left(\left(\Sigma^{-1} - \Psi^{-1}\right)(z-\nu) - \Sigma^{-1}(\mu-\nu)\right)^\top \Psi \left(\left(\Sigma^{-1} - \Psi^{-1}\right)(z-\nu) - \Sigma^{-1}(\mu-\nu)\right) \right], && (47)\\
&= \operatorname{tr}\left[\Psi(\Sigma^{-1} - \Psi^{-1})\Psi(\Sigma^{-1} - \Psi^{-1}) \right] + (\nu-\mu)^\top \Sigma^{-1} \Psi \Sigma^{-1} (\nu-\mu), && (48)\\
&= \operatorname{tr}\left[\left(I - \Psi\Sigma^{-1}\right)^2 \right] + (\nu-\mu)^\top \Sigma^{-1} \Psi \Sigma^{-1} (\nu-\mu). && (49)
\end{aligned}$$

□

Corollary A.8 (Relation to KL divergence). Let p and q be multivariate Gaussian distributions with different means but the same covariance matrix. Then $\frac{1}{2}\mathscr{D}(q;p) = \mathrm{KL}(q;p) = \mathrm{KL}(p;q)$.

Proof. Let μ and ν denote, respectively, the means of p and q, and let Σ denote their shared covariance. From the previous result, we find

$$\mathscr{D}(q;p) = (\nu-\mu)^\top \Sigma^{-1} (\nu-\mu). \tag{50}$$

Finally, we recall the standard derivation for these distributions that

$$\begin{aligned}
\mathrm{KL}(q;p) &= \mathbb{E}_q \left[\log \tfrac{q}{p} \right] && (51)\\
&= \tfrac{1}{2}\mathbb{E}_q \left[(z-\nu)^\top \Sigma^{-1} (z-\nu) - (z-\mu)^\top \Sigma^{-1} (z-\mu) \right] && (52)\\
&= \tfrac{1}{2}\mathbb{E}_q \left[((z-\mu) - (\nu-\mu))^\top \Sigma^{-1} ((z-\mu) - (\nu-\mu)) - (z-\mu)^\top \Sigma^{-1} (z-\mu) \right] && (53)\\
&= \tfrac{1}{2}(\nu-\mu)^\top \Sigma^{-1} (\nu-\mu), && (54)
\end{aligned}$$

thus matching the result for $\frac{1}{2}\mathscr{D}(q;p)$. Moreover, we obtain the same result for $\mathrm{KL}(p;q)$ by noting that the above expression is symmetric with respect to the means μ and ν. □

In sum, the score-based divergence $\mathscr{D}(q;p)$ in eq. (34) has several attractive properties as a measure of difference between most smooth distributions p and q with support on all of $\mathbb{R}^D$. First, it is nonnegative and equal to zero if and only if $p = q$. Second, it is invariant to affine reparameterizations of the underlying domain. Third, it behaves intuitively for simple transformations such as exponential tilting and annealing. Fourth, it is normalized such that every base distribution q has the same divergence to (the limiting case of) a uniform distribution. Finally, it reduces to a constant factor of the KL divergence for the special case of two multivariate Gaussians with the same covariance matrix but different means.

B. Quadratic matrix equations

In this appendix we show how to solve the quadratic matrix equation $XUX + X = V$ where U and V are positive semidefinite matrices in $\mathbb{R}^{D\times D}$. We also verify certain properties of these solutions that are needed elsewhere in the paper but that are not immediately obvious. Quadratic matrix equations of this type (and of many generalizations thereof) have been studied for decades (Potter, 1966; Kučera, 1972a;b; Shurbet et al., 1974; Yuan et al., 2021), and our main goal here is to collect the results that we need in their simplest forms. These results are contained in the following four lemmas.

Lemma B.1. Let $U \succeq 0$ and $V \succ 0$, and suppose that $XUX + X = V$. Then a solution to this equation is given by

$$X = 2V\left[I + (I + 4UV)^{\frac{1}{2}}\right]^{-1}. \tag{55}$$

Proof. We start by turning the left side of the equation $XUX + X = V$ into a form that can be easily factored. Multiplying both sides by U, we see that

$$UXUX + UX = UV. \tag{56}$$

The next step is to complete the square by adding $\frac{1}{4}I$ to both sides; in this way, we find that

$$\left(UX + \tfrac{1}{2}I\right)^2 = UV + \tfrac{1}{4}I. \tag{57}$$

Next we claim that the matrix $UV + \frac{1}{4}I$ on the right side of eq. (57) has all positive eigenvalues. To verify this claim, we note that

$$UV + \tfrac{1}{4}I = V^{-\frac{1}{2}}\left(V^{\frac{1}{2}}UV^{\frac{1}{2}} + \tfrac{1}{4}I\right)V^{\frac{1}{2}}. \tag{58}$$

Thus we see that this matrix is similar to (and thus shares all the same eigenvalues as) the positive definite matrix $U^{\frac{1}{2}}VU^{\frac{1}{2}} + \frac{1}{4}I$ in parentheses on the right side of eq. (58). Since the matrix has all positive eigenvalues, it has a unique principal square root, and from eq. (57) it follows that

$$UX = (UV + \tfrac{1}{4}I)^{\frac{1}{2}} - \tfrac{1}{2}I. \tag{59}$$

If the matrix U were of full rank, then we could solve for X by left-multiplying both sides of eq. (59) by its inverse; however, we desire a general solution even in the case that U is not full rank. Thus we proceed in a different way. In particular, we substitute the solution for UX in eq. (59) into the original form of the quadratic matrix equation. In this way we find that

$$\begin{aligned} V &= XUX + X, && (60)\\ &= X(UX + I), && (61)\\ &= X\left[\left(\left(UV + \tfrac{1}{4}I\right)^{\frac{1}{2}} - \tfrac{1}{2}I\right) + I\right], && (62)\\ &= X\left[(UV + \tfrac{1}{4}I)^{\frac{1}{2}} + \tfrac{1}{2}I\right], && (63)\\ &= \tfrac{1}{2}X\left[(4UV + I)^{\frac{1}{2}} + I\right]. && (64) \end{aligned}$$

Finally we note that the matrix in brackets on the right side of eq. (64) has all positive eigenvalues; hence it is invertible, and after right-multiplying eq. (64) by its inverse we obtain the desired solution in eq. (55). □

Lemma B.2. The solution to $XUX + X = V$ in eq. (55) is symmetric and positive definite.

Proof. The key idea of the proof is to simultaneously diagonalize the matrices U and V^{-1} *by congruence*. In particular, let Λ and E be, respectively, the diagonal and orthogonal matrices satisfying

$$V^{\frac{1}{2}}UV^{\frac{1}{2}} = E\Lambda E^{\top}, \tag{65}$$

where $\Lambda \succeq 0$. Now define $C = V^{\frac{1}{2}}E$. It follows that $C^\top V^{-1} C = I$ and $C^\top U C = \Lambda$, showing that C simultaneously diagonalizes V^{-1} and U by congruence. Alternatively, we may use these relations to express U and V in terms of C and Λ as

$$V = CC^\top, \tag{66}$$

$$U = C^{-\top}\Lambda C^{-1}. \tag{67}$$

We now substitute these expressions for U and V into the solution from eq. (55). The following calculation then gives the desired result:

$$X = 2V\left[I + (I + 4UV)^{-\frac{1}{2}}\right]^{-1}, \tag{68}$$

$$= 2CC^\top\left[I + \left(I + 4C^{-\top}\Lambda C^\top\right)^{\frac{1}{2}}\right]^{-1}, \tag{69}$$

$$= 2CC^\top\left[I + \left(C^{-\top}(I + 4\Lambda)C^\top\right)^{\frac{1}{2}}\right]^{-1}, \tag{70}$$

$$= 2CC^\top\left[I + C^{-\top}(I + 4\Lambda)^{\frac{1}{2}}C^\top\right]^{-1}, \tag{71}$$

$$= 2CC^\top\left[C^{-\top}\left(I + (I + 4\Lambda)^{\frac{1}{2}}\right)C^\top\right]^{-1}, \tag{72}$$

$$= 2CC^\top C^{-\top}\left[I + (I + 4\Lambda)^{\frac{1}{2}}\right]^{-1}C^\top, \tag{73}$$

$$= 2C\left[I + (I + 4\Lambda)^{\frac{1}{2}}\right]^{-1}C^\top. \tag{74}$$

Recalling that $\Lambda \succeq 0$, we see that the above expression for X is manifestly symmetric and positive definite. □

Next we consider the cost of computing the solution to $XUX + X = V$ in eq. (55). On the right side of eq. (55) there appear both a matrix square root and a matrix inverse. As written, it therefore costs $O(D^3)$ to compute this solution when U and V are $D \times D$ matrices. However, if U is of very low rank, there is a way to compute this solution much more efficiently. This possibility is demonstrated by the following lemma.

Lemma B.3 (**Low rank solver**). Let $U = QQ^\top$ where $Q \in \mathbb{R}^{D\times K}$. Then the solution in eq. (55), or equivalently in eq. (74), can also be computed as

$$X = V - V^\top Q\left[\tfrac{1}{2}I + \left(Q^\top V Q + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right]^{-2}Q^\top V. \tag{75}$$

Before proving the lemma, we analyze the computational cost to evaluate eq. (75). Note that it costs $\mathcal{O}(KD^2)$ to compute the decomposition $U = QQ^\top$ as well as to form the product $Q^\top V$, while it costs $\mathcal{O}(K^3)$ to invert and take square roots of $K \times K$ matrices. Thus the total cost of eq. (75) is $\mathcal{O}(KD^2 + K^3)$, in comparison to the $\mathcal{O}(D^3)$ cost of eq. (55). This computational cost results in a potentially large savings if $K \ll D$. We now prove the lemma.

Proof. We will show that eq. (75) is equivalent to eq. (74) in the previous lemma. Again we appeal to the existence of an invertible matrix C that simultaneously diagonalizes V^{-1} and U as in eqs. (66–67). If $U = QQ^\top$, then it follows from eq. (67) that

$$Q = C^{-\top}\Lambda^{\frac{1}{2}}R \tag{76}$$

for some orthogonal matrix R. Next we substitute $V = CC^\top$ from eq. (66) and $Q = C^{-\top}\Lambda^{\frac{1}{2}}R$ from eq. (76) in place of

each appearance of V and Q in eq. (75). In this way we find that

$$X = V - V^\top Q \left[\tfrac{1}{2}I + \left(Q^\top V Q + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right]^{-2} Q^\top V, \tag{77}$$

$$= CC^\top - C\Lambda^{\frac{1}{2}} R \left[\tfrac{1}{2}I + \left((R^\top \Lambda^{\frac{1}{2}} C^{-1})(CC^\top)(C^{-\top}\Lambda^{\frac{1}{2}} R) + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right]^{-2} R^\top \Lambda^{\frac{1}{2}} C^\top, \tag{78}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} R \left[\tfrac{1}{2}I + \left(R^\top \Lambda R + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right]^{-2} R^\top \Lambda^{\frac{1}{2}}\right] C^\top, \tag{79}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} R \left[\tfrac{1}{2}I + R^\top \left(\Lambda + \tfrac{1}{4}I\right)^{\frac{1}{2}} R\right]^{-2} R^\top \Lambda^{\frac{1}{2}}\right] C^\top, \tag{80}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} R \left[R^\top \left(\tfrac{1}{2}I + \left(\Lambda + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right) R\right]^{-2} R^\top \Lambda^{\frac{1}{2}}\right] C^\top, \tag{81}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} R \left[R^\top \left(\tfrac{1}{2}I + \left(\Lambda + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right)^2 R\right]^{-1} R^\top \Lambda^{\frac{1}{2}}\right] C^\top, \tag{82}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} R \left[R^\top \left(\tfrac{1}{2}I + \left(\Lambda + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right)^{-2} R\right] R^\top \Lambda^{\frac{1}{2}}\right] C^\top, \tag{83}$$

$$= C\left[I - \Lambda^{\frac{1}{2}} \left(\tfrac{1}{2}I + \left(\Lambda + \tfrac{1}{4}I\right)^{\frac{1}{2}}\right)^{-2} \Lambda^{\frac{1}{2}}\right] C^\top. \tag{84}$$

We now compare the matrices sandwiched between C and $C^\top$ in eqs. (74) and (84). Both of these sandwiched matrices are diagonal, so it is enough to compare their corresponding diagonal elements. Let ν denote one element along the diagonal of Λ. Then starting from eq. (84), we see that

$$1 - \frac{\nu}{\left(\frac{1}{2} + \sqrt{\nu + \frac{1}{4}}\right)^2} = 1 - \frac{4\nu}{(1+\sqrt{4\nu+1})^2} = \frac{(1+\sqrt{4\nu+1})^2 - 4\nu}{(1+\sqrt{4\nu+1})^2} = \frac{2}{1+\sqrt{4\nu+1}}. \tag{85}$$

Comparing the left and right terms in eq. (85), we see that the corresponding elements of diagonal matrices in eqs. (74) and (84) are equal, and we conclude that eqs. (55) and (75) yield the same solution. □

The last lemma in this appendix is one that we will need for the proof of convergence of Algorithm 1 in the limit of infinite batch size. In particular, it is needed to prove the sandwiching inequality in eq. (26).

Lemma B.4 (Monotonicity). Let X, Y, and V be positive-definite matrices satisfying $XTX + X = YUY + Y = V$, where $T \succeq U \succeq 0$. Then $X \preceq Y$.

Proof. The result follows from examining the solutions for X and Y directly. As shorthand, let $S = V^{\frac{1}{2}}$. By Lemma B.1, we have the solutions

$$X = 2S\left[I + (S + 4STS)^{\frac{1}{2}}\right]^{-1} S, \tag{86}$$

$$Y = 2S\left[I + (S + 4SUS)^{\frac{1}{2}}\right]^{-1} S. \tag{87}$$

If $T \succeq U$, then the positive semi-definite ordering is preserved by the following chain of implications:

$$STS \succeq SUS, \tag{88}$$

$$S + 4STS \succeq S + 4SUS, \tag{89}$$

$$(S + 4STS)^{\frac{1}{2}} \succeq (S + 4SUS)^{\frac{1}{2}}, \tag{90}$$

$$I + (S + 4STS)^{\frac{1}{2}} \succeq I + (S + 4SUS)^{\frac{1}{2}}, \tag{91}$$

where in eq. (90) we have used the fact that positive semi-definite orderings are preserved by matrix square roots. Finally, these orderings are reversed by inverse operations, so that

$$\left[I + (S + 4STS)^{\frac{1}{2}}\right]^{-1} \preceq \left[I + (S + 4SUS)^{\frac{1}{2}}\right]^{-1}. \tag{92}$$

It follows from eq. (92) and the solutions in eqs. (86–87) that $X \preceq Y$, thus proving the lemma. □

C. Derivation of batch and match updates

In this appendix we derive the updates in Algorithm 1 for score-based variational inference. The algorithm alternates between two steps—a BATCH step that draws samples from an approximating Gaussian distribution and computes various statistics of these samples, and a MATCH step that uses these statistics to derive an updated Gaussian approximation, one that better matches the scores of the target distribution. We explain each of these steps in turn, and then we review the special case in which they reduce to the previously published updates (Modi et al., 2023) for Gaussian Score Matching (GSM).

C.1. Batch step

At each iteration, Algorithm 1 solves an optimization based on samples drawn from its current Gaussian approximation to the target distribution. Let q_t denote this approximation at the tth iteration, with mean μ_t and covariance Σ_t, and let $z_1, z_2, \ldots, z_B$ denote the B samples that are drawn from this distribution. The algorithm uses these samples to compute a (biased) empirical estimate of the score-based divergence between the target distribution, p, and another Gaussian approximation q with mean μ and covariance Σ. We denote this empirical estimate by

$$\widehat{\mathscr{D}}_{q_t}(q;p) = \frac{1}{B}\sum_{b=1}^{B}\Big\|\nabla \log q(z_b) - \nabla \log p(z_b)\Big\|_{\Sigma}^2. \tag{93}$$

To optimize the Gaussian approximation q that appears in this divergence, it is first necessary to evaluate the sum in eq. (93) over the batch of samples $z_1, z_2, \ldots, z_B$ that have been drawn from q_t.

The BATCH step of Algorithm 1 computes the statistics of these samples that enter into this calculation. Since q is Gaussian, its score at the bth sample is given by $\nabla \log q(z_b) = -\Sigma^{-1}(z_b - \mu)$. As shorthand, let $g_b = \nabla \log p(z_b)$ denote the score of the target distribution at the bth sample. In terms of these scores, the sum in eq. (93) is given by

$$\widehat{\mathscr{D}}_{q_t}(q;p) \;=\; \frac{1}{B}\sum_{b=1}^{B}\Big\| -\Sigma^{-1}(z_b - \mu) - g_b\Big\|_{\Sigma}^2. \tag{94}$$

Next we show that $\widehat{\mathscr{D}}_{q_t}(q,p)$ depends in a simple way on certain first-order and second-order statistics of the samples, and it is precisely these statistics that are computed in the BATCH step. In particular, we compute the following:

$$\overline{z} = \frac{1}{B}\sum_{b=1}^{B} z_b, \quad \overline{g} = \frac{1}{B}\sum_{b=1}^{B} g_b, \quad C = \frac{1}{B}\sum_{b=1}^{B}(z_b - \overline{z})(z_b - \overline{z})^{\top}, \quad \Gamma = \frac{1}{B}\sum_{n=1}^{N}(g_b - \overline{g})(g_b - \overline{g})^{\top}. \tag{95}$$

Note that the first two of these statistics compute the *means* of the samples and scores in the current iteration of the algorithm, while the remaining two compute their *covariance matrices*. With these definitions, we can now express $\widehat{\mathscr{D}}_{q_t}(q,p)$ in an especially revealing form. Proceeding from eq. (94), we have

$$\widehat{\mathscr{D}}_{q_t}(q;p) = \frac{1}{B}\sum_{b=1}^{B}\Big\|(\overline{g} - g_b) + \Sigma^{-1}(\overline{z} - z_b) + \Sigma^{-1}(\mu - \overline{z} - \Sigma\overline{g})\Big\|_{\Sigma}^2, \tag{96}$$

$$= \frac{1}{B}\sum_{b=1}^{B}\Big[\big\|g_b - \overline{g}\big\|_{\Sigma}^2 + \big\|z_b - \overline{z}\big\|_{\Sigma^{-1}}^2 + \big\|\mu - \overline{z} - \Sigma\overline{g}\big\|_{\Sigma^{-1}}^2 + 2(g_b - \overline{g})(z_b - \overline{z})\Big], \tag{97}$$

$$= \text{tr}(\Gamma\Sigma) \;+\; \text{tr}(C\Sigma^{-1}) \;+\; \big\|\mu - \overline{z} - \Sigma\overline{g}\big\|_{\Sigma^{-1}}^2 \;+\; \text{constant}, \tag{98}$$

where in the second line we have exploited that many cross-terms vanish, and in the third line we have appealed to the definitions of C and Γ in eqs. (95). We have also indicated explicitly that the last term in eq. (98) has no dependence on μ and Σ; it is a constant with respect to the approximating distribution q that the algorithm seeks to optimize. This optimization is performed by the MATCH step, to which we turn our attention next.

C.2. Match step

The MATCH step of the algorithm updates the Gaussian approximation of VI to better match the recently sampled scores of the target distribution. The update at the tth iteration is computed as

$$q_{t+1} = \operatorname*{argmin}_{q \in \mathcal{Q}} \left[\mathscr{L}^{\text{BaM}}(q) \right], \tag{99}$$

where $\mathcal{Q}$ is the Gaussian variational family of Section 2 and $\mathscr{L}^{\text{BaM}}(q)$ is an objective function that balances the empirical estimate of the score based divergence in in eq. (98) against a regularizer that controls how far q_{t+1} can move away from q_t. Specifically, the objective function takes the form

$$\mathscr{L}^{\text{BaM}}(q) = \widehat{\mathscr{D}}_{q_t}(q;p) + \tfrac{2}{\lambda_t}\text{KL}(q_t;q), \tag{100}$$

where the regularizing term is proportional to the KL divergence between the Gaussian distributions q_t and q. This KL divergence is in turn given by the standard result

$$\text{KL}(q_t;q) = \tfrac{1}{2}\left[\text{tr}(\Sigma^{-1}\Sigma_t) - \log\frac{|\Sigma_t|}{|\Sigma|} + \|\mu-\mu_t\|^2_{\Sigma^{-1}} - D \right]. \tag{101}$$

From eqs. (98) and (101), we see that this objective function has a complicated coupled dependence on μ and Σ; nevertheless, the optimal values of μ and Σ can be computed in closed form. The rest of this section is devoted to performing this optimization.

First we perform the optimization with respect to the mean μ, which appears quadratically in the objective $\mathscr{L}^{\text{BaM}}$ through the third terms in (98) and (101). Thus we find

$$\frac{\partial \mathscr{L}^{\text{BaM}}}{\partial \mu} = \frac{\partial}{\partial \mu}\left\{ \left\|\mu - \overline{z} - \Sigma\overline{g}\right\|^2_{\Sigma^{-1}} + \frac{1}{\lambda_t}\|\mu-\mu_t\|^2_{\Sigma^{-1}} \right\} = 2\Sigma^{-1}\left[\mu - \overline{z} - \Sigma\overline{g} + \tfrac{1}{\lambda_t}(\mu-\mu_t)\right]. \tag{102}$$

Setting this gradient to zero, we obtain a linear system which can be solved for the updated mean μ_{t+1} in terms of the updated covariance Σ_{t+1}. Specifically we find

$$\boxed{\mu_{t+1} = \frac{\lambda_t}{1+\lambda_t}\left(\overline{z} + \Sigma_{t+1}\overline{g}\right) + \frac{1}{1+\lambda_t}\mu_t,} \tag{103}$$

matching eq. (13) in Section 3 of the paper. As a sanity check, we observe that in the limit of infinite regularization ($\lambda_t \to 0$), the updated mean is equal to the previous mean (with $\mu_{t+1} = \mu_t$), while in the limit of zero regularization ($\lambda_t \to \infty$), the updated mean is equal to precisely the value that zeros its contribution to $\widehat{\mathscr{D}}_{q_t}(q,p)$ in eq. (98).

Next we perform this optimization with respect to the covariance Σ. To simplify our work, we first eliminate the mean μ from the optimization via eq. (103). When the mean is eliminated in this way from eqs. (98) and (101), we find that

$$\widehat{\mathscr{D}}_{q_t}(q;p) = \text{tr}(\Gamma\Sigma) + \text{tr}(C\Sigma^{-1}) + \frac{1}{(1+\lambda_t)^2}\left\|\mu_t - \overline{z} - \Sigma\overline{g}\right\|^2_{\Sigma^{-1}} + \text{constant}, \tag{104}$$

$$\text{KL}(q_t;q) = \tfrac{1}{2}\left[\text{tr}(\Sigma^{-1}\Sigma_t) - \log\frac{|\Sigma_t|}{|\Sigma|} + \frac{\lambda_t^2}{(1+\lambda_t)^2}\left\|\mu_t - \overline{z} - \Sigma\overline{g}\right\|^2_{\Sigma^{-1}} - D \right]. \tag{105}$$

Combining these terms via eq. (100), and dropping additive constants, we obtain an objective function of the covariance matrix Σ alone. We denote this objective function by $\mathscr{M}(\Sigma)$, and it is given by

$$\mathscr{M}(\Sigma) = \text{tr}(\Gamma\Sigma) + \text{tr}\left(\left[C + \frac{1}{\lambda_t}\Sigma_t\right]\Sigma^{-1}\right) + \frac{1}{1+\lambda_t}\left(\left\|\mu_t - \overline{z}\right\|^2_{\Sigma^{-1}} + \left\|\overline{g}\right\|^2_{\Sigma}\right) + \frac{1}{\lambda_t}\log|\Sigma|. \tag{106}$$

All the terms in this objective function can be differentiated with respect to Σ. To minimize $\mathscr{M}(\Sigma)$, we set its total derivative to zero. Doing this, we find that

$$0 = \Gamma + \frac{1}{1+\lambda_t}\overline{g}\,\overline{g}^\top - \Sigma^{-1}\left[C + \frac{1}{\lambda_t}\Sigma_t + \frac{1}{1+\lambda_t}(\mu_t - \overline{z})(\mu_t - \overline{z})^\top\right]\Sigma^{-1} + \frac{1}{\lambda_t}\Sigma^{-1}. \tag{107}$$

The above is a quadratic matrix equation for the inverse covariance matrix Σ^{-1}; multiplying on the left and right by Σ, we can rewrite it as a quadratic matrix equation for Σ. In this way we find that

$$\boxed{\Sigma U \Sigma + \Sigma = V \quad \text{where} \quad \begin{cases} U = \lambda_t \Gamma + \dfrac{\lambda_t}{1+\lambda_t}\overline{g}\,\overline{g}^\top, \\ V = \Sigma_t + \lambda_t C + \dfrac{\lambda_t}{1+\lambda_t}(\mu_t - \overline{z})(\mu_t - \overline{z})^\top, \end{cases}} \tag{108}$$

matching eq. (9) in Section 3 of the paper. The solution to this quadratic matrix equation is given by Lemma B.1, yielding the update rule

$$\boxed{\Sigma_{t+1} = 2V\left[I + (I + 4UV)^{\frac{1}{2}}\right]^{-1}} \tag{109}$$

and matching eq. (12) in Section 3 of the paper. Moreover, this solution is guaranteed to be symmetric and positive definite by Lemma B.2.

C.3. Gaussian score matching as a special case

In this section, we show that the updates for BaM include the updates for GSM (Modi et al., 2023) as a limiting case. In BaM, this limiting case occurs when there is no regularization ($\lambda \to \infty$) and when the batch size is equal to one ($B = 1$). In this case, we show that the updates in eqs. (103) and (108) coincide with those of GSM.

To see this equivalence, we set $B = 1$, and we use z_t and g_t to denote, respectively, the single sample from q_t and its score under p at the tth iteration of BaM. The equivalence arises from a simple intuition: as $\lambda \to \infty$, all the weight in the loss shifts to minimizing the divergence $\widehat{\mathscr{D}}_{q_t}(q;p)$, which is then minimized exactly so that $\widehat{\mathscr{D}}_{q_t}(q;p) = 0$. More formally, in this limit the batch step can be written as

$$\lim_{\lambda\to\infty} \min_{q\in\mathcal{Q}} \left[\widehat{\mathscr{D}}_{q_t}(q;p) + \frac{2}{\lambda_t}\mathrm{KL}(q_t;q)\right] = \min_{q\in\mathcal{Q}}\left[\mathrm{KL}(q_t;q)\right] \text{ such that } \widehat{\mathscr{D}}_{q_t}(q;p) = 0. \tag{110}$$

The divergence term $\widehat{\mathscr{D}}_{q_t}(q;p)$ only vanishes when the scores match exactly; thus the above can be re-written as

$$\min_{q\in\mathcal{Q}}\left[\mathrm{KL}(q_t;q)\right] \text{ such that } \nabla \log q(z_t) = \nabla \log p(z_t), \tag{111}$$

which is exactly the variational formulation of the GSM method (Modi et al., 2023)

We can also make this equivalence more precise by studying the resulting update. Indeed, the batch statistics in eq. (95) simplify in this setting: namely, we have $\overline{z} = z_t$ and $\overline{g} = g_t$ (because there is only one sample) and $C = \Gamma = 0$ (because the batch has no variance). Next we take the limit $\lambda_t \to \infty$ in eq. (108). In this limit we find that

$$U = g_t g_t^\top, \tag{112}$$

$$V = \Sigma_t + (\mu_t - z_t)(\mu_t - z_t)^\top, \tag{113}$$

so that the covariance is updated by solving the quadratic matrix equation

$$\Sigma_{t+1} g_t g_t^\top \Sigma_{t+1} + \Sigma_{t+1} = \Sigma_t + (\mu_t - z_t)(\mu_t - z_t)^\top. \tag{114}$$

Similarly, taking the limit $\lambda_t \to \infty$ in eq. (103), we see that the mean is updated as

$$\mu_{t+1} = \Sigma_{t+1} g_t + z_t. \tag{115}$$

These BaM updates coincide exactly with the updates for GSM: specifically, eqs. (114) and (115) here are identical to eqs. (42) and (23) in Modi et al. (2023).

D. Proof of convergence

In this appendix we provide full details for the proof of convergence in Theorem 3.1. We repeat equations freely from earlier parts of the paper when it helps to make the appendix more self-contained. Recall that the target distribution in this setting is assumed to be Gaussian with mean μ_* and covariance Σ_*; in addition, we measure the normalized errors at the tth iteration by

$$\varepsilon_t = \Sigma_*^{-\frac{1}{2}}(\mu_t - \mu_*), \tag{116}$$

$$\Delta_t = \Sigma^{-\frac{1}{2}}\Sigma_t\Sigma^{-\frac{1}{2}} - I. \tag{117}$$

If the mean and covariance iterates of Algorithm 1 converge to those of the target distribution, then equivalently the norms of these errors must converge to zero. Many of our intermediate results are expressed in terms of the matrices

$$J_t = \Sigma_*^{-\frac{1}{2}}\Sigma_t\Sigma_*^{-\frac{1}{2}}, \tag{118}$$

which from eq. (117) we can also write as $J_t = I + \Delta_t$. For convenience we restate the theorem in section D.1; our main result is that in the limit of an infinite batch size, the norms of the errors in eqs. (116–117) decay exponentially to zero with rates that we can bound from below.

The rest of the appendix is organized according to the major steps of the proof as sketched in section 3.2. In section D.2, we examine the statistics that are computed by Algorithm 1 when the target distribution is Gaussian and the number of batch samples goes to infinity. In section D.3, we derive the recursions that are satisfied for the normalized mean ε_t and covariance J_t in this limit. In section D.4, we derive a sandwiching inequality for positive-definite matrices that arise in the analysis of these recursions. In section D.5, we use the sandwiching inequality to derive upper and lower bounds on the eigenvalues of J_t. In section D.6, we use these eigenvalue bounds to derive how the normalized errors ε_t and Δ_t decay from one iteration to the next. In section D.7, we use induction on these results to derive the final bounds on the errors in eqs. (121–122), thus proving the theorem. In the more technical sections of the appendix, we sometimes require intermediate results that digress from the main flow of the argument; to avoid too many digressions, we collect the proofs for all of these intermediate results in section D.8.

D.1. Main result

Recall that our main result is that as $B \to \infty$, the spectral norms of the normalized mean and covariance errors in decay exponentially to zero with rates that we can bound from below.

Theorem D.1 (**Restatement of Theorem 3.1**). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$ in Algorithm 1, and let $\alpha > 0$ denote the minimum eigenvalue of the matrix $\Sigma_*^{-\frac{1}{2}}\Sigma_0\Sigma_*^{-\frac{1}{2}}$. For any fixed level of regularization $\lambda > 0$, define

$$\beta := \min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right), \tag{119}$$

$$\delta := \frac{\lambda\beta}{1+\lambda}, \tag{120}$$

where $\beta \in (0, 1]$ measures the quality of initialization and $\delta \in (0, 1)$ denotes a rate of decay. Then with probability 1 in the limit of infinite batch size ($B \to \infty$), and for all $t \geq 0$, the normalized errors in eqs. (116–117) satisfy

$$\|\varepsilon_t\| \leq (1-\delta)^t\|\varepsilon_0\|, \tag{121}$$

$$\|\Delta_t\| \leq (1-\delta)^t\|\Delta_0\| \;+\; t(1-\delta)^{t-1}\|\varepsilon_0\|^2. \tag{122}$$

We emphasize that the theorem holds under very general conditions: it is true no matter how the variational parameters are initialized (assuming only that they are finite and that the initial covariance estimate is not singular), and it is true for any fixed degree of regularization $\lambda > 0$. Notably, the value of λ is *not* required to be inversely proportional to the largest (but a priori unknown) eigenvalue of some Hessian matrix, an assumption that is typically needed to prove the convergence of most

gradient-based methods. This stability with respect to hyperparameters is a well-known property of proximal algorithms, one that has been previously observed beyond the setting of variational inference in this paper.

Finally we note that the bounds in eqs. (121–122) can be tightened with more elaborate bookkeeping and also extended to updates that use varying levels of regularization $\{\lambda_t\}_{t=0}^{\infty}$ at different iterations of the algorithm. At various points in what follows, we indicate how to strengthen the results of the theorem along these lines. Throughout this section, we use the matrix norm $\|\cdot\|$ to denote the spectral norm, and we use the notation $\nu_{\min}(J)$ and $\nu_{\max}(J)$ to denote the minimum and maximum eigenvalues of a matrix J.

D.2. Infinite batch limit

The first step of the proof is analyze how the statistics computed at each iteration of Algorithm 1 simplify in the infinite batch limit ($B \to \infty$). Let q_t denote the Gaussian variational approximation at the tth iteration of the algorithm, let $z_b \sim \mathcal{N}(\mu_t, \Sigma_t)$ denote the bth sample from this distribution, and let $g_b = \nabla \log p(z_b)$ denote the corresponding score of the target distribution p at this sample. Recall that step 5 of Algorithm 1 computes the following batch statistics:

$$\overline{z}_B = \frac{1}{B}\sum_{b=1}^{B} z_b, \qquad C_B = \frac{1}{B}\sum_{b=1}^{B}(z_b - \overline{z}_B)(z_b - \overline{z}_B)^\top, \tag{123}$$

$$\overline{g}_B = \frac{1}{B}\sum_{b=1}^{B} g_b, \qquad \Gamma_B = \frac{1}{B}\sum_{b=1}^{B}(g_b - \overline{g}_B)(g_b - \overline{g}_B)^\top, \tag{124}$$

Here we use the subscript on these averages to explicitly indicate the batch size. (Also, to avoid an excess of indices, we do not explicitly indicate the iteration t of the algorithm.) These statistics simplify considerably when the target distribution is multivariate Gaussian and the number of batch samples goes to infinity. In particular, we obtain the following result.

Lemma D.2 (Infinite batch limit). Suppose $p = \mathcal{N}(\mu_*, \Sigma_*)$. Then with probability 1, as the number of batch samples goes to infinity ($B \to \infty$), the statistics in eqs. (123–124) tend to

$$\lim_{B\to\infty} \overline{z}_B = \mu_t, \tag{125}$$

$$\lim_{B\to\infty} C_B = \Sigma_t, \tag{126}$$

$$\lim_{B\to\infty} \overline{g}_B = \Sigma_*^{-1}(\mu_* - \mu_t), \tag{127}$$

$$\lim_{B\to\infty} \Gamma_B = \Sigma_*^{-1}\Sigma_t\Sigma_*^{-1}. \tag{128}$$

Proof. The first two of these limits follow directly from the strong law of large numbers. In particular, for the sample mean in eq. (123), we have with probability 1 that

$$\lim_{B\to\infty} \overline{z}_B = \lim_{B\to\infty}\left[\frac{1}{B}\sum_{b=1}^{B} z_b\right] = \int z\, q_t(dz) = \mu_t, \tag{129}$$

thus yielding eq. (125). Likewise for the sample covariance in eq. (123), we have with probability 1 that

$$\lim_{B\to\infty} C_B = \lim_{B\to\infty}\left[\frac{1}{B}\sum_{b=1}^{B}(z_b - \overline{z}_B)(z_b - \overline{z}_B)^\top\right] = \int (z-\mu_t)(z-\mu_t)^\top q_t(dz) = \Sigma_t, \tag{130}$$

thus yielding eq. (126). Next we consider the infinite batch limits for $\overline{g}_B$ and Γ_B, in eq. (124), involving the scores of the target distribution. Note that if this target distribution is multivariate Gaussian, with $p = \mathcal{N}(\mu_*, \Sigma_*)$, then we have

$$g_b = \nabla \log p(z_b) = \Sigma_*^{-1}(\mu_* - z_b), \tag{131}$$

showing that the score g_b is a linear function of z_b. Thus the infinite batch limits $\overline{g}_B$ and Γ_B follow directly from those for $\overline{z}_B$ and C_B. In particular, combining eq. (131) with the calculation in eq. (129), we see that

$$\lim_{B\to\infty} \overline{g}_B = \lim_{B\to\infty} \left[\frac{1}{B}\sum_{b=1}^{B} g_b\right] = \lim_{B\to\infty} \left[\Sigma_*^{-1}(\mu_* - \overline{z}_B)\right] = \Sigma_*^{-1}(\mu_* - \mu_t) \tag{132}$$

for the mean of the scores in this limit, thus yielding eq. (127). Likewise, by the same reasoning, we see that

$$\lim_{B\to\infty} \Gamma_B = \lim_{B\to\infty} \left[\frac{1}{B}\sum_{b=1}^{B}(g_b - \overline{g}_B)(g_b - \overline{g}_B)^\top\right] = \lim_{B\to\infty} \Sigma_*^{-1} C_B \Sigma_*^{-1} = \Sigma_*^{-1}\Sigma_t\Sigma_*^{-1} \tag{133}$$

for the covariance of the scores in this limit, thus yielding eq. (128). This proves the lemma. □

D.3. Recursions for ε_t and J_t

Next we use Lemma D.2 to derive recursions for the normalized error ε_t in eq. (116) and the normalized covariance J_t in eq. (118). Both follow directly from our previous results.

Proposition D.3 (Recursion for ε_t). Suppose $p = \mathcal{N}(\mu_*, \Sigma_*)$, and let $B \to \infty$ in Algorithm 1. Then with probability 1, the normalized error at the $(t+1)^{\text{th}}$ iteration of satisfies

$$\varepsilon_{t+1} = \left[I - \frac{\lambda_t}{1+\lambda_t} J_{t+1}\right] \varepsilon_t. \tag{134}$$

Proof. Consider the update for the variational mean in step 7 of Algorithm 1. We begin by computing the infinite batch limit of this update. Using the limits for $\overline{z}_B$ and $\overline{g}_B$ from Lemma D.2, we see that

$$\mu_{t+1} = \lim_{B\to\infty} \left[\left(\frac{1}{1+\lambda_t}\right)\mu_t + \left(\frac{\lambda_t}{1+\lambda_t}\right)\left(\Sigma_{t+1}\overline{g}_B + \overline{z}_B\right)\right], \tag{135}$$

$$= \left(\frac{1}{1+\lambda_t}\right)\mu_t + \left(\frac{\lambda_t}{1+\lambda_t}\right)\left(\Sigma_{t+1}\Sigma_*^{-1}(\mu_* - \mu_t) + \mu_t\right), \tag{136}$$

$$= \mu_t + \frac{\lambda_t}{1+\lambda_t}\Sigma_{t+1}\Sigma_*^{-1}(\mu_* - \mu_t). \tag{137}$$

The proposition then follows by substituting eq. (137) into the definition of the normalized error in eq. (116):

$$\varepsilon_{t+1} = \Sigma_*^{-\frac{1}{2}}(\mu_{t+1} - \mu_*), \tag{138}$$

$$= \Sigma_*^{-\frac{1}{2}}\left[\mu_t + \frac{\lambda_t}{1+\lambda_t}\Sigma_{t+1}\Sigma_*^{-1}(\mu_* - \mu_t) - \mu_*\right], \tag{139}$$

$$= \left[I - \frac{\lambda_t}{1+\lambda_t}\Sigma_*^{-\frac{1}{2}}\Sigma_{t+1}\Sigma_*^{-\frac{1}{2}}\right]\Sigma_*^{-\frac{1}{2}}(\mu_t - \mu_*), \tag{140}$$

$$= \left[I - \frac{\lambda_t}{1+\lambda_t} J_{t+1}\right]\varepsilon_t. \tag{141}$$

This proves the proposition, and we note that this recursion takes the same form as eq. (23), in the proof sketch of Theorem 3.1, if a fixed level of regularization is used at each iteration. □

Proposition D.4 (Recursion for J_t). Suppose $p = \mathcal{N}(\mu_*, \Sigma_*)$, and let $B \to \infty$ in Algorithm 1. Then with probability 1, the normalized covariance at the $(t+1)^{\text{th}}$ iteration of satisfies

$$\lambda_t J_{t+1}\left(J_t + \frac{1}{1+\lambda_t}\varepsilon_t\varepsilon_t^\top\right) J_{t+1} + J_{t+1} = (1+\lambda_t) J_t \tag{142}$$

Proof. Consider the quadratic matrix equation, from step 6 of Algorithm 1, that is satisfied by the variational covariance after $t+1$ updates:

$$\Sigma_{t+1} U_B \Sigma_{t+1} + \Sigma_{t+1} = V_B. \tag{143}$$

We begin by computing the infinite batch limit of the matrices, U_B and V_B, that appear in this equation. Starting from eq. (11) for V_B, and using the limits for $\overline{z}_B$ and C_B from Lemma D.2, we see that

$$\lim_{B\to\infty} V_B = \lim_{B\to\infty} \left[\Sigma_t + \lambda_t C_B + \frac{\lambda_t}{1+\lambda_t}(\mu_t - \overline{z}_B)(\mu_t - \overline{z}_B)^\top\right], \tag{144}$$

$$= (1+\lambda_t)\Sigma_t, \tag{145}$$

$$= \Sigma_*^{\frac{1}{2}}\big[(1+\lambda_t)J_t\big]\Sigma_*^{\frac{1}{2}}, \tag{146}$$

where in the last line we have used eq. (118) to re-express the right side in terms of J_t. Likewise, starting from eq. (10) for U_B, and using the limits for $\overline{g}_B$ and Γ_B from Lemma D.2, we see that

$$\lim_{B\to\infty} U_B = \lim_{B\to\infty} \left[\lambda_t \Gamma_B + \frac{\lambda_t}{1+\lambda_t}\,\bar{g}_B\,\bar{g}_B^\top\right] \tag{147}$$

$$= \lambda_t \Sigma_*^{-1}\Sigma_t\Sigma_*^{-1} + \frac{\lambda_t}{1+\lambda_t}\Sigma_*^{-1}(\mu-\mu_t)(\mu-\mu_t)^\top\Sigma_*^{-1} \tag{148}$$

$$= \lambda_t \Sigma_*^{-1}\Sigma_t\Sigma_*^{-1} + \frac{\lambda_t}{1+\lambda_t}\Sigma_*^{-1}(\mu_*-\mu_t)(\mu_*-\mu_t)^\top\Sigma_*^{-1} \tag{149}$$

$$= \lambda_t \Sigma_*^{-\frac{1}{2}}\left(J_t + \frac{1}{1+\lambda_t}\varepsilon_t\varepsilon_t^\top\right)\Sigma_*^{-\frac{1}{2}}, \tag{150}$$

where again in the last line we have used eqs. (116) and (118) to re-express the right side in terms of ε_t and J_t. Next we substitute these limits for U_B and V_B into the quadratic matrix equation in eq. (143). It follows that

$$\lambda_t \Sigma_{t+1}\Sigma_*^{-\frac{1}{2}}\left(J_t + \frac{1}{1+\lambda_t}\varepsilon_t\varepsilon_t^\top\right)\Sigma_*^{-\frac{1}{2}}\Sigma_{t+1} + \Sigma_{t+1} = \Sigma_*^{\frac{1}{2}}\big[(1+\lambda_t)J_t\big]\Sigma_*^{\frac{1}{2}}. \tag{151}$$

Finally, we obtain the recursion in eq. (142) by left and right multiplying eq. (151) by $\Sigma_*^{-\frac{1}{2}}$ and again making the substitution $J_{t+1} = \Sigma_*^{-\frac{1}{2}}\Sigma_{t+1}\Sigma_*^{-\frac{1}{2}}$ from eq. (118). □

The proof of convergence in future sections relies on various relaxations to derive the simple error bounds in eqs. (121–122). Before proceeding, it is therefore worth noting the following property of Algorithm 1 that is not apparent from these bounds.

Corollary D.5 (One-step convergence). Suppose $p = \mathcal{N}(\mu_*, \Sigma_*)$, and consider the limit of infinite batch size ($B\to\infty$) in Algorithm 1 followed by the *additional* limit of no regularization ($\lambda_0\to\infty$). In this combined limit, the algorithm converges with probability 1 in one step: i.e., $\lim_{\lambda_0\to\infty}\lim_{B\to\infty}\|\varepsilon_1\| = \lim_{\lambda_0\to\infty}\lim_{B\to\infty}\|\Delta_1\| = 0$.

Proof. Consider the recursion for J_1 given by eq. (142) in the additional limit $\lambda_0 \to \infty$. In this limit one can ignore the terms that are not of leading order in λ_0, and the recursion simplifes to $J_1 J_0 J_1 = J_0$. This equation has only one positive-definite solution given by $J_1 = I$. Next consider the recursion for ε_1 given by eq. (134) in the additional limit $\lambda_0 \to \infty$. In this limit this recursion simplifies to $\varepsilon_1 = (I - J_1)\varepsilon_0$, showing that $\varepsilon_1 = 0$. It follows that $\Sigma_1 = \Sigma$ and $\mu_1 = \mu$, and future updates have no effect. □

D.4. Sandwiching inequality

To complete the proof of convergence for Theorem 3.1, we must show that $\|\varepsilon_t\|\to 0$ and $\|J_t - I\|\to 0$ as $t\to\infty$. We showed in Propositions D.3 and D.4 that ε_t and J_t satisfy simple recursions. However, it is not immediately obvious how to translate these recursions for ε_t and J_t into recursions for $\|\varepsilon_t\|$ and $\|J_t - I\|$. To do so requires additional machinery.

One crucial piece of machinery is the *sandwiching inequality* that we prove in this section. In addition to the normalized covariance matrices $\{J_t\}_{t=0}^{\infty}$, we introduce two sequences of *auxiliary* matrices, $\{H_t\}_{t=1}^{\infty}$ and $\{K_t\}_{t=1}^{\infty}$ satisfying

$$0 \prec H_{t+1} \preceq J_{t+1} \preceq K_{t+1} \tag{152}$$

for all $t \geq 0$; this is what we call the sandwiching inequality. These auxiliary matrices are defined by the recursions

$$\lambda_t H_{t+1} \left(J_t + \frac{1}{1+\lambda_t} \|\varepsilon_t\|^2 I \right) H_{t+1} + H_{t+1} = (1+\lambda_t) J_t, \tag{153}$$

$$\lambda_t K_{t+1} J_t K_{t+1} + K_{t+1} = (1+\lambda_t) J_t. \tag{154}$$

We invite the reader to scrutinize the differences between these recursions for H_{t+1} and K_{t+1} and the one for J_{t+1} eq. (142). Note that in eq. (154), defining K_{t+1}, we have dropped the term in eq. (142) involving the outer-product $\varepsilon_t \varepsilon_t^\top$, while in eq. (153), defining H_{t+1}, we have replaced this term by a scalar multiple of the identity matrix. As we show later, these auxiliary recursions are easier to analyze because the matrices H_{t+1} and K_{t+1} (unlike J_{t+1}) share the same eigenvectors as J_t. Later we will exploit this fact to bound their eigenvalues as well as the errors $\|J_{t+1} - I\|$.

In this section we show that the recursions for H_{t+1} and K_{t+1} in eqs. (153–154) imply the sandwiching inequality in eq. (152). As we shall see, the sandwiching inequality follows mainly from the monotonicity property of these quadratic matrix equations proven in Lemma B.4.

Proposition D.6 (Sandwiching inequality). Let $\Sigma_0 \succ 0$ and $\lambda_t > 0$ for all $t \geq 0$. Also, let $\{\varepsilon_t\}_{t=1}^{\infty}$, $\{J_t\}_{t=1}^{\infty}$, $\{H_t\}_{t=1}^{\infty}$, and $\{K_t\}_{t=1}^{\infty}$ be defined, respectively, by the recursions in eqs. (134), (142), and (153–154). Then for all $t \geq 0$ we have

$$0 \prec H_{t+1} \preceq J_{t+1} \preceq K_{t+1}. \tag{155}$$

Proof. We prove the orderings in the proposition from left to right. Since $\Sigma_0 \succ 0$, it follows from eq. (118) that $J_0 \succ 0$, and Lemma B.2 ensures for the recursion in eq. (142) that $J_{t+1} \succ 0$ for all $t \geq 0$. Likewise, since $J_t \succ 0$ for all $t \geq 0$, Lemma B.2 ensures for the recursion in eq. (153) that $H_{t+1} \succ 0$ for all $t \geq 0$. This proves the first ordering in the proposition. To prove the remaining orderings, we note that for all vectors ε_t,

$$\lambda_t J_t \preceq \lambda_t \left(J_t + \frac{1}{1+\lambda_t} \varepsilon_t \varepsilon_t^\top \right) \preceq \lambda_t \left(J_t + \frac{1}{1+\lambda_t} \|\varepsilon_t\|^2 I \right). \tag{156}$$

We now apply Lemma B.4 to the quadratic matrix equations that define the recursions for H_{t+1}, J_{t+1}, and K_{t+1}. From the first ordering in eq. (156), and for the recursions for J_{t+1} and K_{t+1} in eqs. (142) and (154), Lemma B.4 ensures that $J_{t+1} \preceq K_{t+1}$. Likewise, from the second ordering in eq. (156), and for the recursions for J_{t+1} and H_{t+1} in eqs. (142) and (153), Lemma B.4 ensures that $H_{t+1} \preceq J_{t+1}$. □

D.5. Eigenvalue bounds

The sandwiching inequality in the previous section provides a powerful tool for analyzing the eigenvalues of the normalized covariance matrices $\{J_t\}_{t=1}^{\infty}$. As shown in the following lemma, much of this power lies in the fact that the matrices J_t, H_{t+1}, and K_{t+1} are jointly diagonalizable.

Lemma D.7 (Joint diagonalizability). Let $\lambda_t > 0$ for all $t \geq 0$, and let $\{\varepsilon_t\}_{t=1}^{\infty}$, $\{J_t\}_{t=1}^{\infty}$, $\{K_t\}_{t=1}^{\infty}$, and $\{H_t\}_{t=1}^{\infty}$ be defined, respectively, by the recursions in eqs. (134), (142), and (153–154). Then for all $t \geq 0$ we have the following:

(i) H_{t+1} and K_{t+1} share the same eigenvectors as J_t.

(ii) Each eigenvalue ν_J of J_t determines a corresponding eigenvalue ν_H of H_{t+1} and a corresponding eigenvalue ν_K of K_{t+1} via the *positive* roots of the quadratic equations

$$\lambda_t \left(\nu_J + \frac{\|\varepsilon_t\|^2}{1+\lambda_t} \right) \nu_H^2 + \nu_H = (1+\lambda_t) \nu_J, \tag{157}$$

$$\lambda_t \nu_J \nu_K^2 + \nu_K = (1+\lambda_t) \nu_J. \tag{158}$$

Proof. Write $J_t = Q\Lambda_J Q^\top$, where Q is the orthogonal matrix storing the eigenvectors of J_t and Λ_J is the *diagonal* matrix storing its eigenvalues. Now define the matrices

$$\Lambda_H = Q^\top H_{t+1} Q, \tag{159}$$

$$\Lambda_K = Q^\top K_{t+1} Q. \tag{160}$$

We will prove that J_t, H_{t+1}, and K_{t+1} share the same eigenvectors as J_t by showing that the matrices Λ_H and Λ_K are also diagonal. We start by multiplying eqs. (153–154) on the left by $Q^\top$ and on the right by Q. In this way we find

$$\lambda_t \Lambda_H \left(\Lambda_J + \frac{1}{1+\lambda_t} \|\varepsilon_t\|^2 I \right) \Lambda_H + \Lambda_H = (1+\lambda_t)\Lambda_J, \tag{161}$$

$$\lambda_t \Lambda_K \Lambda_J \Lambda_K + \Lambda_K = (1+\lambda_t)\Lambda_J. \tag{162}$$

Since Λ_J is diagonal, we see from eqs. (161–162) that Λ_H and Λ_K also have purely diagonal solutions; this proves the first claim of the lemma. We obtain the scalar equations in eqs. (157–158) by focusing on the corresponding diagonal elements (i.e., eigenvalues) of the matrices Λ_H, Λ_J, and Λ_K in eqs. (161–162); this proves the second claim of the lemma. □

To prove the convergence of Algorithm 1, we will also need upper and lower bounds on eigenvalues of the normalized covariance matrices. The next lemma provides these bounds.

Lemma D.8 (**Bounds on eigenvalues of J_{t+1}**). Let $\lambda_t > 0$ for all $t \geq 0$, and let $\{\varepsilon_t\}_{t=1}^\infty$, $\{J_t\}_{t=1}^\infty$, $\{K_t\}_{t=1}^\infty$, and $\{H_t\}_{t=1}^\infty$ be defined, respectively, by the recursions in eqs. (134), (142), and (153–154). Then for all $t \geq 0$, the largest and smallest eigenvalues of J_{t+1} satisfy

$$\nu_{\max}(J_{t+1}) \leq \sqrt{\frac{1+\lambda_t}{\lambda_t}}, \tag{163}$$

$$\nu_{\min}(J_{t+1}) \geq \min\left(\nu_{\min}(J_t), \frac{1+\lambda_t}{1+\lambda_t+\|\varepsilon_t\|^2} \right). \tag{164}$$

Proof. We will prove these bounds using the sandwiching inequality. We start by proving an upper bound on $\nu_{\max}(K_{t+1})$. Recall from Lemma D.7 that each eigenvalue ν_K of K_{t+1} is determined by a corresponding eigenvalue ν_J of J_t via the positive root of the quadratic equation in eq. (158). Rewriting this equation, we see that

$$\nu_K^2 = \frac{1+\lambda_t}{\lambda_t} - \frac{\nu_K}{\lambda_t \nu_J} \leq \frac{1+\lambda_t}{\lambda_t}, \tag{165}$$

showing that every eigenvalue of K_{t+1} must be less than $\sqrt{\frac{1+\lambda_t}{\lambda_t}}$. Now from the sandwiching inequality, we know that $J_{t+1} \preceq K_{t+1}$, from which it follows that $\nu_{\max}(J_{t+1}) \leq \nu_{\max}(K_{t+1})$. Combining these observations, we have shown

$$\nu_{\max}(J_{t+1}) \leq \nu_{\max}(K_{t+1}) \leq \sqrt{\frac{1+\lambda_t}{\lambda_t}}, \tag{166}$$

which proves the first claim of the lemma. Next we prove a lower bound on $\nu_{\min}(H_{t+1})$. Again, recall from Lemma D.7 that each eigenvalue ν_H of H_{t+1} is determined by a corresponding eigenvalue ν_J of J_t via the positive root of the quadratic equation in eq. (157). We restate this equation here for convenience:

$$\lambda_t \left(\nu_J + \frac{\|\varepsilon_t\|^2}{1+\lambda_t} \right) \nu_H^2 + \nu_H = (1+\lambda_t)\nu_J$$

We now exploit two key properties of this equation, both of which are proven in Lemma D.13. Specifically, Lemma D.13 states that if ν_H is computed from the positive root of this equation, then ν_H is a *monotonically increasing function* of ν_J, and it also satisfies the *lower bound*

$$\nu_H \geq \min\left(\nu_J, \frac{1+\lambda_t}{1+\lambda_t+\|\varepsilon_t\|^2} \right). \tag{167}$$

We can combine these properties to derive a lower bound on the smallest eigenvalue of H_{t+1}; namely, it must be the case that

$$\nu_{\min}(H_{t+1}) \geq \min\left(\nu_{\min}(J_t), \frac{1+\lambda_t}{1+\lambda_t+\|\varepsilon_t\|^2}\right). \tag{168}$$

Now again from the sandwiching inequality, we know that $J_{t+1} \succeq H_{t+1}$, from which it follows that $\nu_{\min}(J_{t+1}) \geq \nu_{\min}(H_{t+1})$. Combining this observation with eq. (168), we see that

$$\nu_{\min}(J_{t+1}) \geq \nu_{\min}(H_{t+1}) \geq \min\left(\nu_{\min}(J_t), \frac{1+\lambda_t}{1+\lambda_t+\|\varepsilon_t\|^2}\right), \tag{169}$$

which proves the second claim of the lemma. □

D.6. Recursions for $\|\varepsilon_t\|$ and $\|\Delta_t\|$

In this section, we analyze how the errors $\|\varepsilon_t\|$ and $\|\Delta_t\|$ evolve from one iteration of Algorithm 1 to the next. These *per-iteration* results are the cornerstone of the proof of convergence in the infinite batch limit.

Proposition D.9 (Decay of $\|\varepsilon_t\|$). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$. Then for Algorithm 1 in the limit of infinite batch size ($B \to \infty$), the normalized errors in eq. (116) of the variational mean *strictly decrease* from one iteration to the next: i.e., $\|\varepsilon_{t+1}\| < \|\varepsilon_t\|$. More precisely, they satisfy

$$\|\varepsilon_{t+1}\| \leq \left(1 - \frac{\lambda_t}{1+\lambda_t}\nu_{\min}(J_{t+1})\right)\|\varepsilon_t\|, \tag{170}$$

where the multiplier in parentheses on the right side is strictly less than one.

Proof. Recall from Proposition D.3 that the normalized errors in the variational mean satisfy the recursion

$$\varepsilon_{t+1} = \left[I - \frac{\lambda_t}{1+\lambda_t}J_t\right]\varepsilon_t. \tag{171}$$

Taking norms and applying the sub-multiplicative property of the spectral norm, we have

$$\|\varepsilon_{t+1}\| \leq \left\|I - \frac{\lambda_t}{1+\lambda_t}J_{t+1}\right\|\|\varepsilon_t\|. \tag{172}$$

Consider the matrix norm that appears on the right side of eq. (172). By Lemma D.8, and specifically eq. (163) which gives the ordering $J_{t+1} \preceq \sqrt{\frac{1+\lambda_t}{\lambda_t}}I$, it follows that

$$I - \frac{\lambda_t}{1+\lambda_t}J_{t+1} \succeq \left(1 - \sqrt{\frac{\lambda_t}{1+\lambda_t}}\right)I \succ 0. \tag{173}$$

Thus the spectral norm of this matrix is strictly greater than zero and determined by the minimum eigenvalue of J_{t+1}. In particular, we have

$$\left\|I - \frac{\lambda_t}{1+\lambda_t}J_t\right\| = 1 - \frac{\lambda_t}{1+\lambda_t}\nu_{\min}(J_{t+1}), \tag{174}$$

and the proposition is proved by substituting eq. (174) into eq. (172). □

Proposition D.10 (Decay of $\|\Delta_t\|$). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$. Then for Algorithm 1 in the limit of infinite batch size ($B \to \infty$), the normalized errors in eq. (117) of the variational covariance satisfy

$$\|\Delta_{t+1}\| \leq \|\varepsilon_t\|^2 + \frac{1}{1+\lambda_t\nu_{\min}(J_t)}\|\Delta_t\|. \tag{175}$$

Proof. We start by applying the triangle inequality and the sandwiching inequality:

$$\|\Delta_{t+1}\| = \|J_{t+1} - I\|, \tag{176}$$
$$\leq \|J_{t+1} - K_{t+1}\| \; + \; \|K_{t+1} - I\|, \tag{177}$$
$$\leq \|H_{t+1} - K_{t+1}\| \; + \; \|K_{t+1} - I\|. \tag{178}$$

Already from these inequalities we can see the main outlines of the result in eq. (175). Clearly, the first term in eq. (178) must vanish when $\|\varepsilon_t\| = 0$ because the auxiliary matrices H_{t+1} and K_{t+1}, defined in eqs. (153–154), are equal when $\varepsilon_t = 0$. Likewise, the second term in eq. (178) must vanish when $\|\Delta_t\| = 0$, or equivalently when $J_t = I$, because in this case eq. (154) is also solved by $K_{t+1} = I$.

First we consider the left term in eq. (178). Recall from Lemma D.7 that the matrices H_{t+1} and K_{t+1} share the same eigenvectors; thus the spectral norm $\|H_{t+1} - K_{t+1}\|$ is equal to the largest gap between their corresponding eigenvalues. Also recall from eqs. (157–158) of Lemma D.7 that these corresponding eigenvalues ν_H and ν_K are determined by the positive roots of the quadratic equations

$$\lambda_t \left(\nu_J + \frac{\|\varepsilon_t\|^2}{1+\lambda_t} \right) \nu_H^2 + \nu_H = (1+\lambda_t)\nu_J, \tag{179}$$
$$\lambda_t \nu_J \nu_K^2 + \nu_K = (1+\lambda_t)\nu_J, \tag{180}$$

where ν_J is their (jointly) corresponding eigenvalue of J_t. Since these two equations agree when $\|\varepsilon_t\|^2 = 0$, it is clear that $|\nu_H - \nu_K| \to 0$ as $\|\varepsilon_t\| \to 0$. More precisely, as we show in Lemma D.14 of section D.8, it is the case that

$$|\nu_H - \nu_K| \leq \|\varepsilon_t\|^2. \tag{181}$$

(Specifically, this is property (v) of Lemma D.14.) It follows in turn from this property that

$$\|H_{t+1} - K_{t+1}\| \; \leq \; \|\varepsilon_t\|^2. \tag{182}$$

We have thus bounded the left term in eq. (178) by a quantity that, via Proposition D.9, is decaying geometrically to zero with the number of iterations of the algorithm.

Next we focus on the right term in eq. (178). The spectral norm $\|K_{t+1} - I\|$ is equal to the largest gap between any eigenvalue of K_{t+1} and the value of 1 (i.e., the value of all eigenvalues of I). Recall from eq. (158) of Lemma D.7 that each eigenvalue ν_J of J_t determines a corresponding eigenvalue ν_K of K_{t+1} via the positive root of the quadratic equation

$$\lambda_t \nu_J \nu_K^2 + \nu_K = (1+\lambda_t)\nu_J. \tag{183}$$

This correspondence has an important *contracting* property that eigenvalues of J_t not equal to one are mapped to eigenvalues of K_{t+1} that are closer to one. In particular, as we show in Lemma D.13 of section D.8, it is the case that

$$|\nu_K - 1| \leq \frac{1}{1+\lambda_t \nu_J} |\nu_J - 1|. \tag{184}$$

(Specifically, this is property (vii) of Lemma D.13.) It follows in turn from this property that

$$\|K_{t+1} - I\| \; \leq \; \frac{1}{1+\lambda_t \nu_{\min}(J_t)} \|J_t - I\|. \tag{185}$$

Finally, the proposition is proved by substituting eq. (182) and eq. (185) into eq. (178). □

The results of Proposition D.9 and Proposition D.10 could be used to further analyze the convergence of Algorithm 1 when different levels of regularization λ_t are used at each iteration. By specializing to a fixed level of regularization, however, we obtain the especially interpretable results of eqs. (19–20) in the proof sketch of Theorem 3.1. To prove these results, we need one further lemma.

Lemma D.11 (**Bound on $\nu_{\min}(J_t)$**). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$ in Algorithm 1, and let $\alpha > 0$ denote the minimum eigenvalue of the matrix $\Sigma_*^{-\frac{1}{2}} \Sigma_0 \Sigma_*^{-\frac{1}{2}}$. Then in the limit of infinite batch size ($B \to \infty$), and for any fixed level of regularization $\lambda > 0$, we have for all $t \geq 0$ that

$$\nu_{\min}(J_t) \geq \min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right). \tag{186}$$

Proof. We prove the result by induction. Note that $\nu_{\min}(J_0) = \nu_{\min}\left(\Sigma_*^{-\frac{1}{2}} \Sigma_0 \Sigma_*^{-\frac{1}{2}}\right) = \alpha$, so that eq. (186) holds for $t = 0$. Now assume that the result holds for some iteration $t > 0$. Then

$$\nu_{\min}(J_{t+1}) \geq \min\left(\nu_{\min}(J_t), \frac{1+\lambda}{1+\lambda+\|\varepsilon_t\|^2}\right), \tag{187}$$

$$\geq \min\left(\min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right), \frac{1+\lambda}{1+\lambda+\|\varepsilon_t\|^2}\right), \tag{188}$$

$$= \min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right), \tag{189}$$

where the first inequality is given by eq. (164) of Lemma D.8, the second inequality follows from the inductive hypothesis, and the final equality holds because $\|\varepsilon_t\| < \|\varepsilon_0\|$ from Proposition D.9. □

Note how the bound in eq. (186) depends on α and $\|\varepsilon_0\|$, both of which reflect the quality of initialization. In particular, when $\alpha \ll 1$, the initial covariance is close to singular, and when $\|\varepsilon_0\|$ is large, the initial mean is a poor estimate. Both these qualities of initialization play a role in the next result.

Corollary D.12 (**Rates of decay for $\|\varepsilon_t\|$ and $\|\Delta_t\|$**). Suppose that $p = \mathcal{N}(\mu_*, \Sigma_*)$ and let $\alpha > 0$ denote the minimum eigenvalue of the matrix $\Sigma_*^{-\frac{1}{2}} \Sigma_0 \Sigma_*^{-\frac{1}{2}}$. Also, for any fixed level of regularization $\lambda > 0$, define

$$\beta = \min\left(\alpha, \frac{1+\lambda}{1+\lambda+\|\varepsilon_0\|^2}\right), \tag{190}$$

$$\delta = \frac{\lambda\beta}{1+\lambda}, \tag{191}$$

where $\beta \in (0, 1]$ measures the quality of initialization and $\delta \in (0, 1)$ measures a rate of decay. Then in the limit of infinite batch size ($B \to \infty$), the normalized errors in eqs. (116–117) satisfy

$$\|\varepsilon_{t+1}\|^2 \leq (1-\delta)^2 \|\varepsilon_t\|^2, \tag{192}$$

$$\|\Delta_{t+1}\| \leq (1-\delta)\|\Delta_t\| + \|\varepsilon_t\|^2. \tag{193}$$

Proof. The results follow from the previous ones in this section. In particular, from Proposition D.9 and the previous lemma, we see that

$$\|\varepsilon_{t+1}\| \leq \left(1 - \frac{\lambda}{1+\lambda}\nu_{\min}(J_{t+1})\right)\|\varepsilon_t\| \leq \left(1 - \frac{\lambda\beta}{1+\lambda}\right)\|\varepsilon_t\| = (1-\delta)\|\varepsilon_t\|. \tag{194}$$

Likewise, from Proposition D.10 and the previous lemma, we see that

$$\|\Delta_{t+1}\| \leq \|\varepsilon_t\|^2 + \frac{1}{1+\lambda\nu_{\min}(J_t)} \|\Delta_t\|, \tag{195}$$

$$\leq \|\varepsilon_t\|^2 + \frac{1}{1+\lambda\beta} \|\Delta_t\|, \tag{196}$$

$$= \|\varepsilon_t\|^2 + \left(1 - \frac{\lambda\beta}{1+\lambda\beta}\right) \|\Delta_t\|, \tag{197}$$

$$\leq \|\varepsilon_t\|^2 + \left(1 - \frac{\lambda\beta}{1+\lambda}\right) \|\Delta_t\|, \tag{198}$$

$$= \|\varepsilon_t\|^2 + (1-\delta) \|\Delta_t\|. \tag{199}$$

□

D.7. Induction

From the previous corollary we can at last give a simple proof of Theorem 3.1. It should also be clear that tighter bounds can be derived, and differing levels of regularization accommodated, if we instead proceed from the more general bounds in Propositions D.9 and D.10.

Proof of Theorem 3.1. We start from eqs. (192–193) of Corollary D.12 and proceed by induction. At iteration $t=0$, we see from these equations that

$$\|\varepsilon_1\| \leq (1-\delta)\|\varepsilon_0\|, \tag{200}$$

$$\|\Delta_1\| \leq (1-\delta)\|\Delta_0\| + \|\varepsilon_0\|^2. \tag{201}$$

The above agree with eqs. (17–18) at iteration $t=0$ and therefore establish the base case of the induction. Next we assume the inductive hypothesis that eqs. (17–18) are true at some iteration $t-1$. Then again, appealing to eqs. (192–193) of Corollary D.12, we see that

$$\|\varepsilon_t\| \leq (1-\delta)\|\varepsilon_{t-1}\|, \tag{202}$$

$$\leq (1-\delta)(1-\delta)^{t-1}\|\varepsilon_0\|, \tag{203}$$

$$= (1-\delta)^t\|\varepsilon_0\|, \tag{204}$$

$$\|\Delta_t\| \leq (1-\delta)\|\Delta_{t-1}\| + \|\varepsilon_{t-1}\|^2, \tag{205}$$

$$\leq (1-\delta)\Big[(1-\delta)^{t-1}\|\Delta_0\| + (t-1)(1-\delta)^{t-2}\|\varepsilon_0\|^2\Big] + (1-\delta)^{2(t-1)}\|\varepsilon_0\|^2, \tag{206}$$

$$= (1-\delta)^t\|\Delta_0\| + \Big[(t-1)(1-\delta)^{t-1} + (1-\delta)^{2t-2}\Big]\|\varepsilon_0\|^2, \tag{207}$$

$$\leq (1-\delta)^t\|\Delta_0\| + \Big[(t-1)(1-\delta)^{t-1} + (1-\delta)^{t-1}\Big]\|\varepsilon_0\|^2, \tag{208}$$

$$= (1-\delta)^t\|\Delta_0\| + t(1-\delta)^{t-1}\|\varepsilon_0\|^2. \tag{209}$$

This proves the theorem. □

D.8. Supporting lemmas

In this section we collect a number of lemmas whose results are needed throughout this appendix but whose proofs digress from the main flow of the argument.

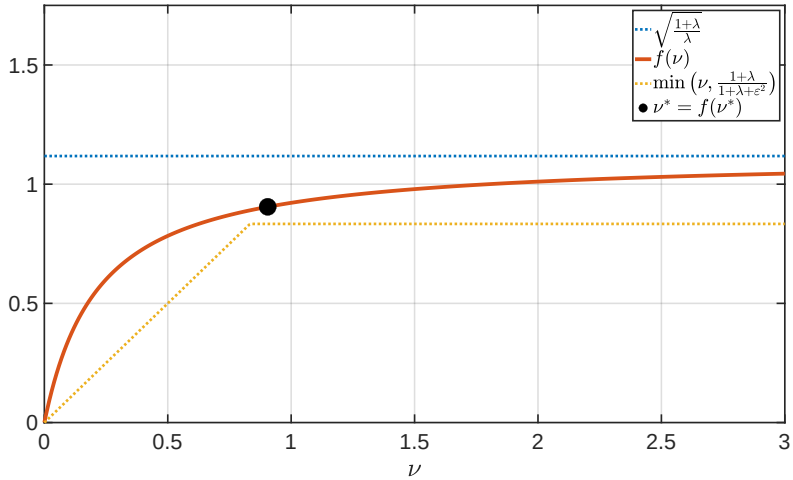


Figure D.1: Plot of the function f in eq. (211), as well as its fixed point and upper and lower bounds from Lemma D.13, with $\lambda = 4$ and $\varepsilon^2 = 1$.

Lemma D.13. Let $\lambda > 0$ and $\varepsilon^2 \geq 0$, and let $f : \mathbb{R}_+ \to \mathbb{R}_+$ be the function defined implicitly as follows: if $\nu > 0$ and $\xi = f(\nu)$, then ξ is equal to the *positive* root of the quadratic equation

$$\lambda \left(\nu + \frac{\varepsilon^2}{1+\lambda} \right) \xi^2 + \xi - (1+\lambda)\nu \;=\; 0. \tag{210}$$

Then f has the following properties:

(i) f is monotonically increasing on $(0, \infty)$.
(ii) $f(\nu) < \sqrt{\frac{1+\lambda}{\lambda}}$ for all $\nu > 0$.
(iii) f has a unique fixed point $\nu^* = f(\nu^*)$.
(iv) $f(\nu) \geq \nu^*$ for all $\nu \geq \nu^*$.
(v) $f(\nu) > \nu$ for all $\nu \in (0, \nu^*)$.
(vi) $f(\nu) \geq \min\left(\nu, \frac{1+\lambda}{1+\lambda+\varepsilon^2}\right)$ for all $\nu > 0$.
(vii) If $\varepsilon^2 = 0$, then $|\nu - 1| \geq (1+\lambda\nu)|f(\nu) - 1|$ for all $\nu > 0$.

Before proving the lemma, we note that it is straightforward to solve the quadratic equation in eq. (210). Doing so, we find

$$f(\nu) = \frac{-1 + \sqrt{1 + 4\lambda(1+\lambda)\nu^2 + 4\lambda\varepsilon^2\nu}}{2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)}. \tag{211}$$

In most aspects, this explicit form for f is less useful than the implicit one given in the statement of the lemma. However, eq. (211) is useful for visualizing properties (i)-(vi), and Fig. D.1 shows a plot of $f(\nu)$ with $\lambda = 4$ and $\varepsilon^2 = 1$. We now prove the lemma.

Proof. Let $\nu > 0$. To prove property (i) that f is monotonically increasing, it suffices to show $f'(\nu) > 0$. Differentiating eq. (210) with respect to ν, we find that

$$\lambda\xi^2 + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi f'(\nu) + f'(\nu) - (1+\lambda) = 0, \tag{212}$$

where $\xi = f(\nu)$. To proceed, we re-arrange terms to isolate $f'(\nu)$ on the left side and use eq. (210) to remove quadratic powers of ξ. In this way, we find:

$$\left[1 + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi\right] f'(\nu) = 1 + \lambda - \lambda\xi^2, \tag{213}$$

$$= 1 + \lambda - \frac{(1+\lambda)\nu - \xi}{\nu + \frac{\varepsilon^2}{1+\lambda}}, \tag{214}$$

$$= \frac{\xi + \varepsilon^2}{\nu + \frac{\varepsilon^2}{1+\lambda}}. \tag{215}$$

Note that the term in brackets on the left side is strictly positive, as is the term on the right side. It follows that $f'(\nu) > 0$, thus proving property (i). Moreover, since f is monotonically increasing, it follows from eq. (211) that

$$f(\nu) \;<\; \lim_{\omega\to\infty} f(\omega) \;=\; \sqrt{\frac{1+\lambda}{\lambda}}, \tag{216}$$

thus proving property (ii). To prove property (iii), we solve for fixed points of f. Let $\nu^* > 0$ denote a fixed point satisfying $\nu^* = f(\nu^*)$. Then upon setting $\nu = \nu^*$ in eq. (210), we must find that $\xi = \nu^*$ is a solution of the resulting equation, or

$$\lambda\left(\nu^* + \frac{\varepsilon^2}{1+\lambda}\right)\nu^{*2} + \nu^* - (1+\lambda)\nu^* \;=\; 0. \tag{217}$$

Eq. (217) has one root at zero, one negative root, and one positive root, but only the last of these can be a fixed point of f, which is defined over $\mathbb{R}_+$. This fixed point corresponds to the positive root of the quadratic equation:

$$\left(\nu^* + \frac{\varepsilon^2}{1+\lambda}\right)\nu^* \;=\; 1. \tag{218}$$

This proves property (iii). Property (iv) follows easily from properties (i) and (iii): if $\nu \geq \nu^*$, then $f(\nu) \geq f(\nu^*) = \nu^*$, where the inequality holds because f is monotonically increasing and the equality holds because ν^* is a fixed point of f. To prove property (v), suppose that $\nu \in (0, \nu^*)$. Then from eq. (218), it follows that

$$\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\nu \;<\; 1. \tag{219}$$

Now let $\xi = f(\nu)$. Then from eq. (210) and eq. (219), it follows that

$$0 = \nu \cdot 0 \tag{220}$$

$$= \nu\left[\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi^2 + \xi - (1+\lambda)\nu\right], \tag{221}$$

$$= \lambda\nu\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi^2 \;+\; \nu\xi \;-\; (1+\lambda)\nu^2, \tag{222}$$

$$< \lambda\xi^2 \;+\; \nu\xi \;-\; (1+\lambda)\nu^2, \tag{223}$$

$$= (\xi - \nu)(\xi + (1+\lambda)\nu). \tag{224}$$

Since the right factor in eq. (224) is positive, the inequality as a whole can only be satisfied if $\xi > \nu$, or equivalently if $f(\nu) > \nu$, thus proving property (v). To prove property (vi), we observe from eq. (218) that $\nu^* \leq 1$, and from this *upper* bound on ν^*, we re-use eq. (218) to derive the *lower* bound

$$\nu^* = \frac{1}{\nu^* + \frac{\varepsilon^2}{1+\lambda}} \geq \frac{1}{1 + \frac{\varepsilon^2}{1+\lambda}} = \frac{1+\lambda}{1+\lambda+\varepsilon^2}. \tag{225}$$

With this lower bound, we show next that property (vi) follows from properties (iv) and (v). In particular, if $\nu \in (0, \nu^*)$, then from property (v) we have $f(\nu) > \nu$; on the other hand, if $\nu \geq \nu^*$, then from property (iv) and the lower bound in eq. (225), we have $f(\nu) \geq \nu^* \geq \frac{1+\lambda}{1+\lambda+\varepsilon^2}$. But either $\nu \in (0, \nu^*)$ or $\nu \geq \nu^*$, and hence for all $\nu > 0$ we have

$$f(\nu) \geq \min\left(\nu, \frac{1+\lambda}{1+\lambda+\varepsilon^2}\right), \tag{226}$$

which is exactly property (vi). Fig. (D.1) plots the lower and upper bounds on f from properties (ii) and (vi), as well as the fixed point $\nu^* = f(\nu^*)$. Property (vii) considers the special case when $\varepsilon^2 = 0$. In this case, we can also rewrite eq. (210) as

$$\nu - 1 \;=\; \lambda\nu\xi^2 + \xi - \lambda\nu - 1 \;=\; (1 + \lambda\nu + \lambda\nu\xi)(\xi - 1), \tag{227}$$

and taking the absolute values of both sides, we find that

$$|\nu - 1| \;=\; (1 + \lambda\nu + \lambda\nu\xi)|\xi - 1| \;\geq (1 + \lambda\nu)|\xi - 1| \tag{228}$$

for all $\nu > 0$, thus proving property (vii). The meaning of this property becomes more evident upon examining the function's fixed point: note from eq. (218) that $\nu^* = 1$ when $\varepsilon^2 = 0$. Thus property (vii) can alternatively be written as

$$|f(\nu) - \nu^*| \leq \frac{1}{1 + \lambda\nu}|\nu - \nu^*|, \tag{229}$$

showing that the function converges to its fixed point when it is applied in an iterative fashion. □

Lemma D.14. Let $\lambda, \nu > 0$, and let $g : [0, \infty) \to \mathbb{R}_+$ be the function defined implicitly as follows: if $\xi = g(\varepsilon^2)$, then ξ is equal to the *positive* root of the quadratic equation

$$\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi^2 + \xi - (1+\lambda)\nu \;=\; 0. \tag{230}$$

Then g has the following properties:

(i) g is monotonically decreasing on $[0, \infty)$.
(ii) $g(0) < \sqrt{\frac{1+\lambda}{\lambda}}$.
(iii) $g'(0) > -1$.
(iv) g is convex on $[0, \infty)$.
(v) $|g(\varepsilon^2) - g(0)| \leq \varepsilon^2$.

Before proving the lemma, we note that it is straightforward to solve the quadratic equation in eq. (230). Doing so, we find

$$g(\varepsilon^2) = \frac{-1 + \sqrt{1 + 4\lambda(1 + \lambda)\nu^2 + 4\lambda\varepsilon^2\nu}}{2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)}. \tag{231}$$

This explicit formula for g is not needed for the proof of the lemma. However, eq. (231) is useful for visualizing properties (i)-(ii), and Fig. D.2 shows several plots of $g(\varepsilon^2)$ for different values of λ and ν. We now prove the lemma.

Proof. To prove property (i) that g is monotonically increasing, it suffices to show $g'(\varepsilon^2) < 0$. Differentiating eq. (230) with respect to ε^2, we find that

$$\frac{\lambda}{1+\lambda}\xi^2 + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi g'(\varepsilon^2) + g'(\varepsilon^2) = 0 \tag{232}$$

where $\xi = g(\varepsilon^2)$, and solving for $g'(\varepsilon)$, we find that

$$g'(\varepsilon^2) = -\frac{\lambda\xi^2}{(1+\lambda)(1+2\lambda\nu\xi) + 2\lambda\varepsilon^2\xi} < 0, \tag{233}$$

which proves property (i). To prove property (ii), let $\xi_0 = g(0)$ denote the positive root of eq. (230) when $\varepsilon^2 = 0$. Then this root satisfies

$$\xi_0^2 = \frac{1+\lambda}{\lambda} - \frac{\xi_0}{\lambda\nu} < \frac{1+\lambda}{\lambda}, \tag{234}$$

from which the result follows. Moreover, it follows from eqs. (233–234) that

$$g'(0) = -\frac{\lambda\xi_0^2}{(1+\lambda)(1 + 2\lambda\nu\xi_0)} > -\frac{\lambda\xi_0^2}{1+\lambda} > -\frac{\lambda}{1+\lambda}\frac{1+\lambda}{\lambda} = -1, \tag{235}$$

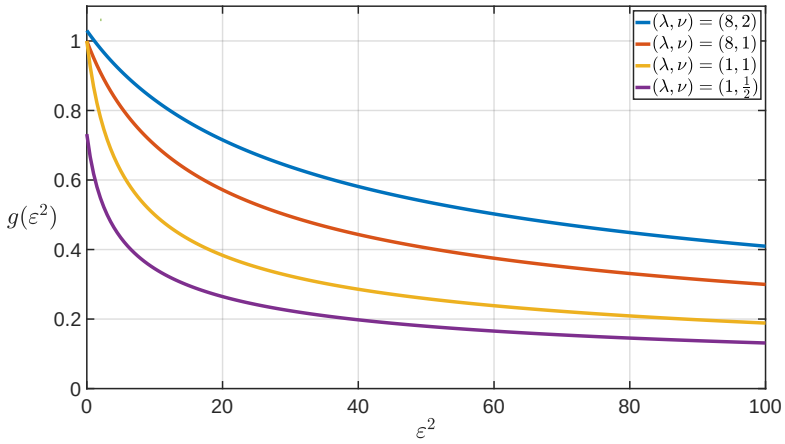


Figure D.2: Plot of the function g in Lemma D.14 and eq. (231) for several different values of λ and ν.

thus proving property (iii). To prove property (iv) that g is convex, it suffices to show $g''(\varepsilon^2) > 0$. Differentiating eq. (232) with respect to ε^2, we find that

$$\frac{4\lambda\xi}{1+\lambda}\, g'(\varepsilon^2) + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\left(\xi g''(\varepsilon^2) + g'(\varepsilon^2)^2\right) + g''(\varepsilon^2) = 0. \tag{236}$$

To proceed, we re-arrange terms to isolate $g''(\varepsilon^2)$ on the left side and use eq. (232) to re-express the term on the right. In this way, we find:

$$\left[1 + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi\right] g''(\varepsilon^2) = -\frac{4\lambda\xi}{1+\lambda} g'(\varepsilon^2) - 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right) g'(\varepsilon^2)^2, \tag{237}$$

$$= -\frac{g'(\varepsilon^2)}{\xi}\left[\frac{4\lambda\xi^2}{1+\lambda} + 2\lambda\left(\nu + \frac{\varepsilon^2}{1+\lambda}\right)\xi g'(\varepsilon^2)\right], \tag{238}$$

$$= -\frac{g'(\varepsilon^2)}{\xi}\left[\frac{4\lambda\xi^2}{1+\lambda} - \frac{\lambda\xi^2}{1+\lambda} - g'(\varepsilon^2)\right], \tag{239}$$

$$= -\frac{g'(\varepsilon^2)}{\xi}\left[\frac{3\lambda\xi^2}{1+\lambda} - g'(\varepsilon^2)\right]. \tag{240}$$

Note that the term in brackets on the left side is strictly positive, and because g is monotonically decreasing, with $g'(\varepsilon^2) < 0$, so is the term on the right. It follows that $g''(\varepsilon^2) > 0$, thus proving property (iv). Finally, to prove property (v), we combine the results that g is monotonically decreasing, that its derivative at zero is greater than -1, and that it is convex:

$$|g(\varepsilon^2) - g(0)| \;=\; g(0) - g(\varepsilon^2) \;\leq\; g(0) - (g(0) + g'(0)\varepsilon^2) = -g'(0)\varepsilon^2 \;\leq\; \varepsilon^2. \tag{241}$$

□

E. Additional experiments and details

E.1. Implementation of baselines

In Algorithm 2, we describe the version of ADVI implemented in the experiments. In particular, we use ADAM as the optimizer for updating the variational parameters. We also implemented an alternate version of ADVI using the score-based divergence and the Fisher divergence in place of the (negative) ELBO loss. In Algorithm 3, we also describe the implementation of the GSM algorithm (Modi et al., 2023).

Algorithm 2 Implementation of ADVI

Input: Iterations T, batch size B, unnormalized target $\tilde{p}$, learning rate $\lambda_t > 0$, initial variational mean $\mu_0 \in \mathbb{R}^D$, initial variational covariance $\Sigma_0 \in \mathbb{S}^D_{++}$
for $t = 0, \ldots, T-1$ **do**
  Sample $z_1, \ldots, z_B \sim q_t = \mathcal{N}(\mu_t, \Sigma_t)$
  Compute stochastic estimate of the (negative) ELBO

$$\mathcal{L}^{(t)}_{\text{ELBO}}(z_{1:B}) = -\sum_{b=1}^{B} \log(\tilde{p}(z_b) - \log q_t(z_b))$$

  Update variational parameters $w_t := (\mu_t, \Sigma_t)$ with gradient

$$w_{t+1} = w_t - \lambda_t \nabla_w \mathcal{L}^{(t)}_{\text{ELBO}}(z_{1:B}) \qquad \textit{\# Our implementation uses the ADAM update.}$$

end for
Output: variational parameters μ_T, Σ_T

Algorithm 3 Implementation of GSM

Input: Iterations T, batch size B, unnormalized target $\tilde{p}$, initial variational mean $\mu_0 \in \mathbb{R}^D$, initial variational covariance $\Sigma_0 \in \mathbb{S}^D_{++}$
for $t = 0, \ldots, T-1$ **do**
  Sample $z_1, \ldots, z_B \sim q_t = \mathcal{N}(\mu_t, \Sigma_t)$
  for $b = 1, \ldots, B$ **do**
    Compute the score of the sample $s_b = \nabla_z \log(\tilde{p}(z_b))$
    Calculate intermediate quantities

$$\varepsilon_b = \Sigma_t s_b - \mu_t + z_b, \qquad \text{and solve } \rho(1+\rho) = s_b^\top \Sigma_t s_b + \left[(\mu_t - z_b)^\top s_b\right]^2 \text{ for } \rho > 0$$

    Estimate the update for mean and covariance

$$\delta\mu_b = \tfrac{1}{1+\rho}\left[I - \tfrac{(\mu_t - z_b)s_b^\top}{1+\rho+(\mu_t - z_b)^\top s_b}\right]\varepsilon_b$$
$$\delta\Sigma_b = (\mu_t - z_b)(\mu_t - z_b)^\top - (\tilde{\mu}_b - z_b)(\tilde{\mu}_b - z_b)^\top, \qquad \text{where } \tilde{\mu}_b = \mu_t + \delta\mu_b$$

  end for
  Update variational mean and covariance

$$\mu_{t+1} = \mu_t + \tfrac{1}{B}\sum_{b=1}^{B}\delta\mu_b, \qquad \Sigma_{t+1} = \Sigma_t + \tfrac{1}{B}\sum_{b=1}^{B}\delta\Sigma_b$$

end for
Output: variational parameters μ_T, Σ_T

E.2. Wallclock timings

In the main paper, we report the number of gradient evaluations as a measure of the cost of the algorithm. While the complete cost is not captured by the number of gradient evaluations alone, here we show that the computational cost of the algorithms are dominated by gradient evaluations, and so number of gradient evaluations is a good proxy of the computational cost. We additionally note that all work with full covariance matrices make a basic assumption that $\mathcal{O}(D^2)$ is not prohibitive because there are $\mathcal{O}(D^2)$ parameters in the model itself. While the BaM update (when $B \geq D$) takes $\mathcal{O}(D^3)$ computation per iteration, in this setting, $\mathcal{O}(D^3)$ is not generally regarded as prohibitive in models where there are $\mathcal{O}(D^2)$ parameters to estimate.

In Figure E.1, we plot the wallclock timings for Gaussian targets of increasing dimension, where $D = 4, 16, 64, 128, 256$. We observe that for dimensions 64 and under, all methods have similar timings; for the larger dimensions, we observe that the low-rank BaM solver has a similar timing. All experiments in the paper fit into the lower-dimensional regime or the low-rank regime, with the exception of the deep generative models application, which includes larger batch sizes. Thus, for the lower-dimensional regime and the low-rank examples, we report the number of gradient evaluations as the primary measure of cost; the cost per iteration for the mini-batch regime is $\mathcal{O}(D^2B + B^3)$. For the deep generative model example, we additionally report in Figure E.7 the wallclock timings. We note that the wallclock timings themselves are heavily dependent on implementation and JIT-compilation details and hardware.

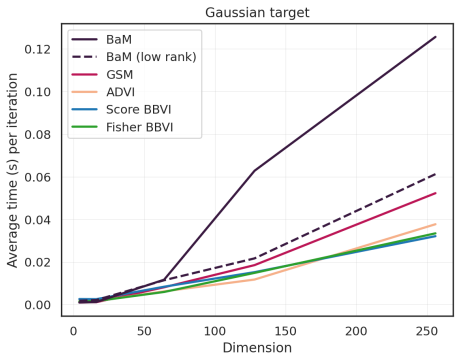


Figure E.1: Wallclock timings for the Gaussian targets example.

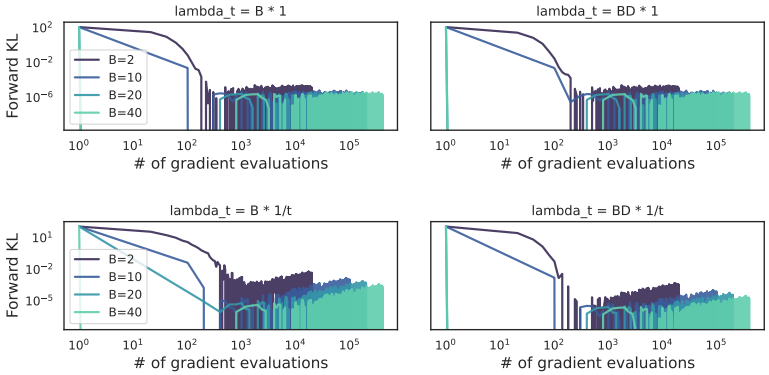


Figure E.2: Gaussian target, $D = 16$

E.3. Gaussian target

Each target distribution was generated randomly; here the covariance was constructed by generating a $D \times D$ matrix A and computing $\Sigma_* = AA^\top$.

For all experiments, the algorithms were initialized with $\mu_0 \sim \text{uniform}[0, 0.1]$ and $\Sigma_0 = I$. In Figure E.3, we report the results for the reverse KL divergence. We observe largely the same conclusions as with the forward KL divergence presented in Section 5.

In addition, we evaluated BaM with a number of different schedules for the learning rates: $\lambda_t = B, BD, \frac{B}{t+1}, \frac{BD}{t+1}$. We show one such example for $D = 16$ in Figure E.2, where each figure represents a particular choice of λ_t, and where each line is the mean over 10 runs. For the constant learning rate, the lines for $B = 20, 40$ are on top of each other. Here we observe that the constant learning rates perform the best for Gaussian targets. For the gradient-based methods (ADVI, Score, Fisher), the learning rate was set by choosing the best value over a grid search. For ADVI and Fisher, the selected learning rate was 0.01. For Score, a different learning rate was selected for each dimension $D = 4, 16, 64, 256$: $[0.01, 0.005, 0.001, 0.001]$.

E.4. Non-Gaussian target

Here we again consider the sinh-arcsinh distribution with $D = 10$, where we vary the skew and tails. We present the reverse KL results in Figure E.4.

All algorithms were initialized with a random initial mean μ_0 and $\Sigma_0 = I$. In Figure E.5, we present several alternative plots showing the forward and reverse KL divergence when varying the learning rate. We investigate the performance for different schedules corresponding to $\lambda_t = BD, \frac{BD}{\sqrt{t+1}}, \frac{BD}{(t+1)}$, and we varied the batch size $B = 2, 5, 10, 20, 40$. Unlike for Gaussian targets, we found that constant λ_t did not perform as well as those with a varying schedule. In particular, we found that $\lambda_t = \frac{BD}{t+1}$ typically converges faster than the other schedule.

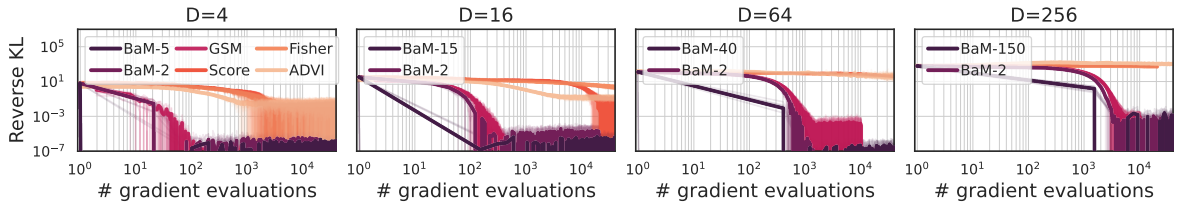


Figure E.3: Gaussian targets of increasing dimension. Solid curves indicate the mean over 10 runs (transparent curves). ADVI, Score, Fisher, and GSM use batch size of 2. The batch size for BaM is given in the legend.

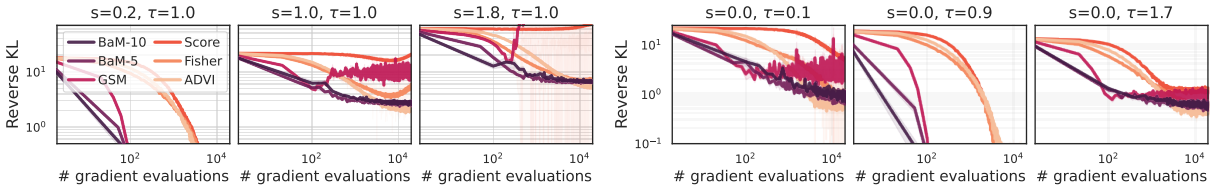


Figure E.4: Non-Gaussian targets constructed using the sinh-arcsinh distribution, varying the skew s and the tail weight t. ADVI and GSM use a batch size of $B = 5$.

For the gradient-based methods (ADVI, Score, Fisher), a grid search was run over the learning rate for ADAM. The final selected learning rates were 0.02 for ADVI and 0.05 for Fisher. For Score, more tuning was required: for the targets with fixed tails $\tau = 1$ and varying skew $s = 0.2, 1, 1.8$, the learning rates $[0.01, 0.001, 0.001]$ were used, and for the targets with fixed skew $s = 0$ and varying tails $\tau = 0.1, 0.9, 1.7$, the learning rates $[0.001, 0.01, 0.01]$, respectively. We note that for the score-based divergence, several of the highly skewed targets led to divergence (with respect to the grid search) on most of the random seeds that were run.

E.5. Posteriordb models

In Bayesian posterior inference applications, it is common to measure the relative mean error and the relative standard deviation error (Welandawe et al., 2022):

$$\text{relative mean error} = \left\| \frac{\mu - \hat{\mu}}{\sigma} \right\|_2, \quad \text{relative SD error} = \left\| \frac{\sigma - \hat{\sigma}}{\sigma} \right\|_2, \tag{242}$$

where $\hat{\mu}, \hat{\sigma}$ are computed from the variational distribution, and μ, σ are the posterior mean and standard deviation. We estimated the posterior mean and standard deviation using the reference samples from HMC.

In the evaluation, all algorithms were initialized with $\mu_0 \sim \text{uniform}[0, 0.1]$ and $\Sigma_0 = I$. The results for the relative mean error are presented in Section 5. In Figure E.6, we present the results for the relative SD error. Here we typically observe the same trends as for the mean, except in the hierarchical example, in which BaM learns the mean quickly but converges to a larger relative SD error. However, the low error of GSM suggests that more robust tuning of the learning rate may lead to better performance with BaM.

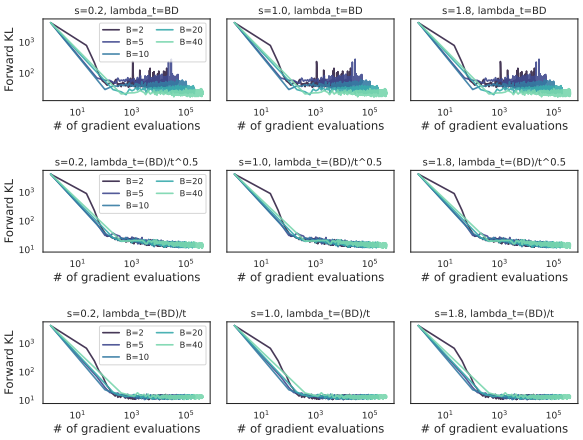
E.6. Deep learning model

In Figure E.7, we present the results from the main paper but with wallclock times on the x-axis. We arrive at similar conclusions: here BaM with $B = 300$ converges the fastest compared to GSM and ADVI using any batch size.

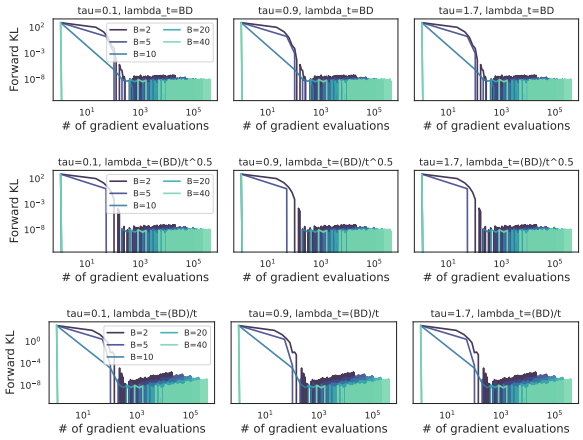
We provide additional details for the experiment conducted in Section 5.3. We first pre-train the neural network $\Omega(\cdot, \hat{\theta})$ (the "decoder") using variational expectation-maximization. That is, $\hat{\theta}$ maximizes the marginal likelihood $p(\{x_n\}_{n=1}^N \mid \theta)$, where $\{x_n\}_{n=1}^N$ denotes the training set. The marginalization step is performed using an approximation

$$q(z_n|x_n) \approx p(z_n|x_n, \theta),$$

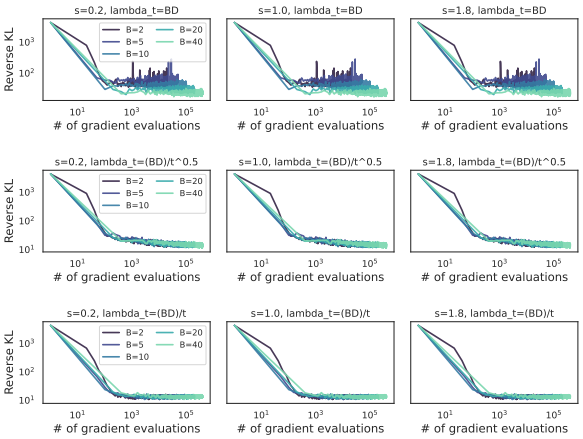
obtained with amortized variational inference. In details, we optimize the ELBO over the family of factorized Gaussians and learn an inference neural network (the "encoder") that maps x_n to the parameters of $q(z_n \mid x_n)$. This procedure is standard



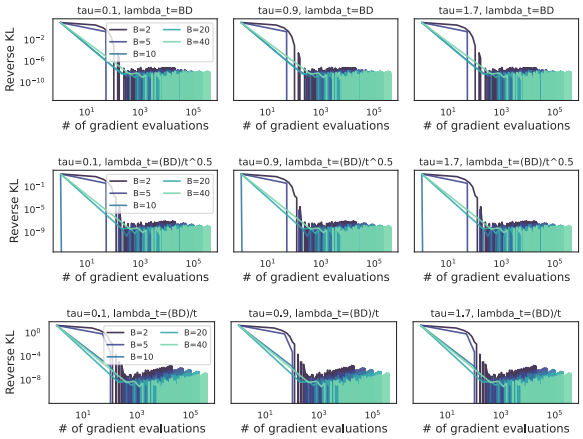
(a) Forward KL: varying λ_t and skew



(b) Forward KL: varying λ_t and tails



(c) Reverse KL: varying λ_t and skew



(d) Reverse KL: varying λ_t and tails

Figure E.5: Non-Gaussian target, $D = 10$. Panels (a) and (b) show the forward KL, and panels (c) and (d) show the reverse KL.

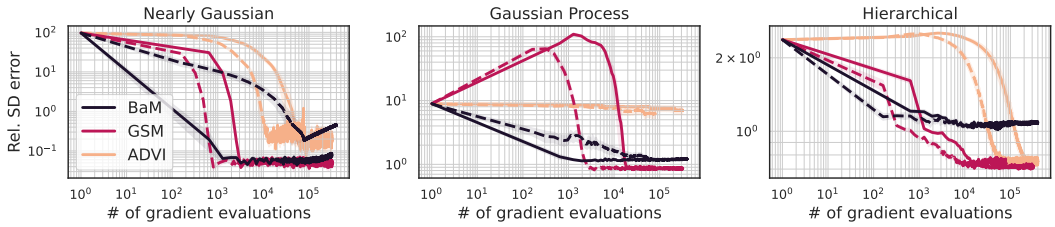


Figure E.6: Posterior inference in Bayesian models measured by the relative standard deviation error. The curves denote the mean over 5 runs, and shaded regions denote their standard error. Solid curves ($B = 32$) correspond to larger batch sizes than the dashed curves ($B = 8$).

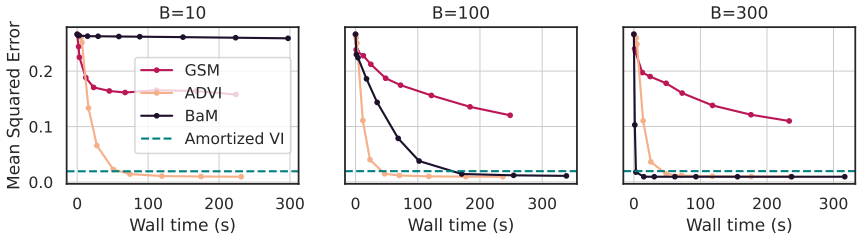


Figure E.7: Image reconstruction error when the posterior mean of z' is fed into the generative neural network. The x-axis denotes the wallclock time in seconds.

for training a VAE (Kingma & Welling, 2014; Rezende et al., 2014; Tomczak, 2022). For the decoder and the encoder, we use a convolution network with 5 layers. The optimization is performed over 100 epochs, after which the ELBO converges (Figure E.8).

For the estimation of the posterior on a new observation, we draw an image x' from the test set. All VI algorithms are initialized at a standard Gaussian. For ADVI and BaM, we conduct a pilot experiment of 100 iterations and select the learning rate that achieves the lowest MSE for each batch size ($B = 10, 100, 300$). For ADVI, we consistently find the best learning rate to be $\ell = 0.02$ (after searching $\ell = 0.001, 0.01, 0.02, 0.05$). For BaM, we find that different learning rates work better for different batch sizes:

- $B = 10$, $\lambda = 0.1$ selected from $\lambda = 0.01, 0.1, 0.2, 10$.
- $B = 100$, $\lambda = 50$ selected from $\lambda = 2, 20, 50, 100, 200$.
- $B = 300$, $\lambda = 7500$ selected from $\lambda = 1000, 5000, 7500, 10000$.

For $B = 300$, all candidate learning rates achieve the minimal MSE (since BaM converges in less than 100 iterations), and so we pick the one that yields the fastest convergence.

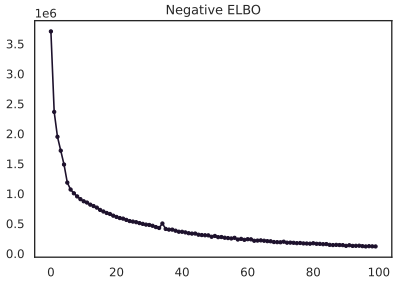


Figure E.8: ELBO for variational autoencoder over 100 epochs