Variational Inference with Gaussian Score Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

Variational inference (VI) is a method to approximate the computationally intractable posterior distributions that arise in Bayesian statistics. Typically, VI fits a simple parametric distribution to be close to the target posterior, minimizing an appropriate objective such as the evidence lower bound (ELBO). In this work, we present a new approach to VI. Our method is based on the principle of score matching, that if two distributions are equal then their score functions (i.e., gradients of the log density) are equal at every point on their support. With this principle, we develop score matching VI, an iterative algorithm that seeks to match the scores between the variational approximation and the exact posterior. At each iteration, score matching VI solves an inner optimization, one that minimally adjusts the current variational estimate to match the scores at a newly sampled value of the latent variables. We show that when the variational family is a Gaussian, this inner optimization enjoys a closed form solution, which we call Gaussian score matching VI (GSM-VI). GSM-VI is also a "black box" variational algorithm in that it only requires a differentiable joint distribution, and as such it can be applied to a wide class of models. We compare GSM-VI to black box variational inference (BBVI), which has similar requirements but instead optimizes the ELBO. We first study how GSM-VI behaves as a function of the problem dimensionality, the condition number of the target covariance matrix (when the target is Gaussian), and the degree of mismatch between the approximating and exact posterior distribution. We then study GSM-VI on a collection of real-world Bayesian inference problems from the posteriorDB database of datasets and models. In all of our studies we find that GSM-VI is faster than BBVI, but without sacrificing accuracy. It requires 10-100x fewer gradient evaluations to obtain a comparable quality of approximation.

1 Introduction

This paper is about variational inference for approximate Bayesian computation. Consider a statistical model $p(\boldsymbol{\theta}, \boldsymbol{x})$ of parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ and observations $\boldsymbol{x}$. Bayesian inference aims to infer the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{x})$, which is often intractable to compute. Variational inference is an optimization-based approach to approximate the posterior [4, 16].

The idea behind VI is to approximate the posterior with a member of a *variational family* of distributions $q_{\boldsymbol{w}}(\boldsymbol{\theta})$, parameterized by *variational parameters* $\boldsymbol{w}$ [4, 16]. Specifically, VI methods establish a measure of closeness between $q_{\boldsymbol{w}}(\boldsymbol{\theta})$ and the posterior, and then minimize it with an optimization algorithm. Researchers have explored many aspects of VI, including different objectives [7, 8, 18, 22, 23, 25, 30] and optimization strategies [1, 13, 24].

In its modern form, VI typically minimizes $\mathrm{KL}\left(q_{\boldsymbol{w}}(\theta) || p(\theta \mid x)\right)$ with stochastic optimization, and further satisfies the so-called "black-box" criteria [1, 24, 29]. The resulting black-box VI (BBVI) only

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

requires the practitioner to specify the log joint $\log p(\boldsymbol{\theta}, \boldsymbol{x})$ and (often) its gradient $\nabla_{\boldsymbol{\theta}} \log p(\theta, \boldsymbol{x})$, which for many models can be obtained by automatic differentiation. For these reasons, BBVI has been widely implemented, and it is available in many probabilistic programming systems [3, 19, 27].

In this paper, we propose a new approach to VI. We begin with the principle of *score matching* [14], that when two densities are equal then their gradients are equal as well, and we use this principle to derive a new way to fit a variational distribution to be close to the exact posterior. The result is *score-matching VI*. Rather than explicitly minimize a divergence, score-matching VI iteratively projects the variational distribution onto the exact score matching constraint. This strategy enables a new black-box VI algorithm.

Score-matching VI relies on the same ingredients as reparameterization BBVI [19]—a differentiable variational family and a differentiable log joint—and so it can be as easily incorporated into probabilistic programming systems as well. Further, when the variational family is a Gaussian, score-matching VI is particularly efficient: each iteration is computable in closed form. We call the resulting algorithm Gaussian score matching VI (GSM-VI).

Unlike BBVI, GSM-VI does not rely on stochastic gradient descent (SGD) for its core optimization. Though SGD has the appeal of simplicity, it is also known to require the careful tuning of learning rates. GSM-VI was inspired by a different tradition of constraint-based algorithms for online learning [2, 6, 11, 12, 20]. These algorithms have been extensively developed and analyzed for problems in classification, and under the right conditions, they have been observed to outperform SGD. This paper shows how to extend this constraint-based framework—and the powerful machinery behind it—from the problem of classification to the workhorse of Gaussian VI. The key insight is that score-matching (unlike ELBO maximization) lends itself naturally to a constraint-based formulation.

We empirically compared GSM-VI to reparameterization BBVI on several classes of models, and with both synthetic and real-world data. In general, we found that GSM requires 10-100x fewer gradient evaluations to converge to an equally good approximation. When the exact posterior is Gaussian, we found GSM-VI scales significantly better with respect to dimensionality and is insensitive to the condition number of the target covariance. When the exact posterior is non-Gaussian, we found GSM-VI enjoys faster convergence without sacrificing the quality of the final approximation.

This paper makes the following contributions:

- We introduce *score matching variational inference*, a new black-box approach to fitting $q_{\boldsymbol{w}}(\boldsymbol{\theta})$ to be close to $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. Score matching VI requires no tunable optimization hyperamarameters, to which BBVI can be sensitive.
- When the variational family is Gaussian, we develop *Gaussian score matching variational inference* (GSM-VI). It establishes efficient closed-form iterates for score matching VI.
- We empirically compare GSM-VI to reparameterization BBVI. Across many models and datasets, we found that GSM-VI enjoys faster convergence to an equally good approximation.

We develop score matching VI in Section 2 and study its performance in Section 3.

Related work. Our work introduces a new method for black-box variational inference that relies only on having access to the gradients of the variational distribution and the log joint. GSM-VI has similar goals to automatic-differentiation variational inference (ADVI) [19] and Pathfinder [31], which also fit multivariate Gaussian variational families, but do so by maximizing the ELBO using stochastic optimization. Similar to GSM-VI, the algorithm of ref. [28] also seeks to match the scores of the variational and the target posterior, but it does so by minimizing the L2 loss between them.

A novel aspect of this work is how GSM-VI fits the variational parameters. Rather than minimize a loss function, it aims to solve a set of nonlinear equations. Similar ideas have been pursued in the context of fitting a model to data using empirical risk minimization (ERM). For example, passive agressive (PA) methods [6] and the stochastic polyak stepsize (SPS) are also derived via projections onto sampled nonlinear equations [2, 12, 20]. A probabilistic extension of PA methods is known as confidence-weighted (CW) learning [11]. In this framework, the learner maintains a multivariate Gaussian distribution over the weight vector of a linear classifier. Like CW learning, the second

step of GSM-VI also minimizes a KL divergence between multivariate Gaussians. But it involves a different projection, one of score-matching versus linear classification.

2 Score Matching Variational Inference

Suppose for the moment that the variational family $q_{\boldsymbol{w}}(\boldsymbol{\theta})$ is rich enough to perfectly capture the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. That is, there exists a $\boldsymbol{w}^*$ such that

$$\log\, q_{\boldsymbol{w}^*}(\boldsymbol{\theta}) = \log\, p(\boldsymbol{\theta}|\boldsymbol{x}), \qquad \forall \boldsymbol{\theta}\,. \tag{1}$$

If we could solve Eq. 1 for $\boldsymbol{w}^*$, the resulting variational distribution would be a perfect fit. The challenge is that the posterior on the right side is intractable to compute.

To help, we appeal to score matching [14]. Define the score of a distribution to be the gradient of its log with respect to the variable[1], e.g., $\nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{w}}(\boldsymbol{\theta})$. The principle of score matching is that if two distributions are equal at each point in their support then their score functions are also equal.

To use score matching for VI, we first write the log posterior as the log joint minus the normalizing constant, i.e., the marginal distribution of $\boldsymbol{x}$,

$$\log\, p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \log\, p(\boldsymbol{\theta}, \boldsymbol{x}) - \log\, p(\boldsymbol{x}). \tag{2}$$

With this expression, the principle of score matching leads to the following Lemma.

Lemma 2.1. The parameter $\boldsymbol{w}^*$ satisfies

$$\nabla_{\boldsymbol{\theta}} \log\, q_{\boldsymbol{w}^*}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log\, p(\boldsymbol{\theta}, \boldsymbol{x}), \qquad \forall \boldsymbol{\theta}, \tag{3}$$

if and only if $\boldsymbol{w}^*$ also satisfies Eq. 1.

What is notable about Eq. 3 is that the right side is the gradient of the log joint. Unlike the posterior, the gradient of the log joint is tractable to compute for a large class of probabilistic models. (The proof is in the appendix.)

This lemma motivates a new algorithm, *score matching VI*. The idea is to iteratively refine the variational parameters $\boldsymbol{w}$ to try to solve the system of equations in Eq. 3 as well as possible. At each iteration t, it first samples a new $\boldsymbol{\theta}_t$ from the current variational approximation and then minimally adjusts $\boldsymbol{w}$ to satisfy Eq. 3 for that value of $\boldsymbol{\theta}_t$.

Score matching variational inference

At iteration t:

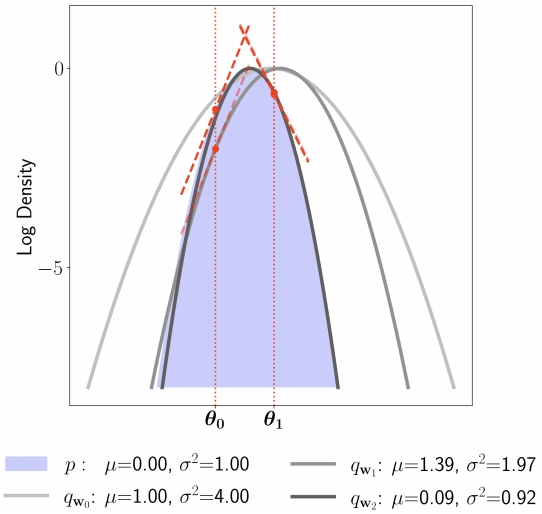
1. Sample $\boldsymbol{\theta}_t \sim q_{\boldsymbol{w}_t}(\boldsymbol{\theta})$.
2. Update the variational parameters:

$$\begin{aligned} \boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}} &\operatorname{KL}\left(q_{\boldsymbol{w}_t}(\boldsymbol{\theta}) \,||\, q_{\boldsymbol{w}}(\boldsymbol{\theta})\right) \\ &\text{such that} \quad \nabla_{\boldsymbol{\theta}}\, \log q_{\boldsymbol{w}}(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}}\, \log p(\boldsymbol{\theta}_t, \boldsymbol{x}). \end{aligned} \tag{4}$$

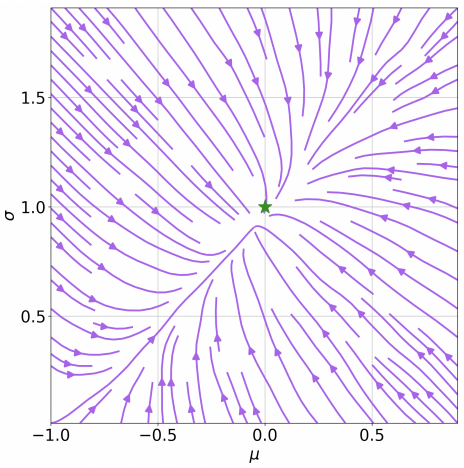
This algorithm for score matching VI was inspired by earlier online algorithms for learning a classifier from a stream of labeled examples. One particularly elegant algorithm in this setting is known as passive-aggressive (PA) learning [6], in which a model is incrementally updated by the minimal amount to classify each example correctly by a large margin. This approach was subsequently extended to a probabilistic setting, known as confidence-weighted (CW) learning [11] in which one minimally updates a *distribution* over classifiers. Our algorithm is similar in that it minimally updates an approximating distribution for VI, but it is different in that enforces constraints for score matching instead of large margin classification.

At a high level, what makes this approach to VI likely to succeed or fail? Certainly it is necessary that there are more variational parameters than elements of the latent variable $\boldsymbol{\theta}$; when this is not the

[1] We make this clear because, in some literature, the score is the gradient with respect to the parameter.



(a) Two iterations of GSM-VI. The log density of the target posterior p is shaded blue; the initial distribution $q_{\boldsymbol{w}_0}$ is light grey; the first update $q_{\boldsymbol{w}_1}$ is medium grey; and the second update $q_{\boldsymbol{w}_2}$ is dark grey.



(b) The vector field of Eq. 4, averaged over 5 independent samples, where $p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \mathcal{N}(0, 1)$. The solution $(\mu, \sigma) = (0, 1)$ is the green star.

case, it may be impossible to satisfy a *single* score matching constraint in Eq. 4. That said, setting the number of variational parameters to be at least as large as the latent variable is standard, as in, e.g., a factorized (or mean-field) variational family. It is also apparent that the algorithm may never converge if the target posterior is not contained in the variational family, or that the the variational approximation collapses to a point mass, which stalls the updates altogether. While we cannot dismiss these possibilities out of hand, we did not see either issue in any of the empirical studies of Section 3.

For more intuition, Figure 1a illustrates the effect of the update in Eq. 4 when both the target and approximating distribution are 1d Gaussian. The target posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ is shaded blue. The plot shows the initial variational distribution $q_{\boldsymbol{w}_0}$ (light grey curve) and its update to $q_{\boldsymbol{w}_1}$ (medium grey) so that the gradient of the updated distribution matches the gradient of the target at the sampled $\boldsymbol{\theta}_0$ (dotted red tangent line). It also shows the update from $\boldsymbol{w}_1$ to $\boldsymbol{w}_2$, now matching the gradient at$\boldsymbol{\theta}_1$. With these two updates, $q_{\boldsymbol{w}_2}$ (dark grey) is very close to the target $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. With this picture in mind, we now develop the details of this algorithm for a widely applicable setting.

Gaussian Score Matching VI. Suppose the variational distribution belongs to a multivariate Gaussian family $q_{\boldsymbol{w}}(\boldsymbol{\theta}) := \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which is a common setting especially in systems for automated approximate inference [1, 19]. One of our main contributions is to show that in this case Eq. 4 has a closed form solution. The solution $\boldsymbol{w}_{t+1} = (\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$ has the following form:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \boldsymbol{A}_t \left(\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t, \boldsymbol{x}) - \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{w}_t}(\boldsymbol{\theta}_t)\right) \tag{5}$$

$$\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t + (\boldsymbol{\mu}_t - \boldsymbol{\theta}_t)(\boldsymbol{\mu}_t - \boldsymbol{\theta}_t)^\top - (\boldsymbol{\mu}_{t+1} - \boldsymbol{\theta}_t)(\boldsymbol{\mu}_{t+1} - \boldsymbol{\theta}_t)^\top \tag{6}$$

where $\boldsymbol{A}_t \in \mathbb{R}^{d \times d}$ is a matrix defined in the theorem below. These exact updates only require the score of the log joint $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \boldsymbol{x})$ and the score of the variational distribution $\nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{w}}(\boldsymbol{\theta})$.

Eqs. 5 and 6 also provide intuition. Consider the approximation at the tth iteration $q_{\boldsymbol{w}_t}$ and the current sample $\boldsymbol{\theta}_t$. First suppose the scores already match at this sample, that is $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t, \boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{w}_t}(\boldsymbol{\theta}_t)$. Then the mean does not change $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t$ and, similarly, the two rank-one terms in the covariance update in Eq. 6 cancel out so $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t$. This shows that when $q_{\boldsymbol{w}_t}(\boldsymbol{\theta}) = p(\boldsymbol{\theta}, \boldsymbol{x})$ for all $\boldsymbol{\theta}$, the method stops. On the other hand, if the scores do not match, then the mean is updated proportionally to the difference between the scores, and the covariance is updated by a rank-two correction. For a one dimensional target $p(\boldsymbol{\theta}, \boldsymbol{x}) = \mathcal{N}(0, 1)$, Figure 1b illustrates the vector field of updates The vector field points to the solution (green star) and, once there, the method stops.

We now formalize this result and give the exact expression for $\boldsymbol{A}_t$.

Algorithm 1: Gaussian Score Matching VI

Input : Initial mean estimate $\boldsymbol{\mu}_0$, initial covariance estimate $\boldsymbol{\Sigma}_0$, target distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$, number of iterations $N \in \mathbb{N}$, batch size $B \in \mathbb{N}$.

Output : Multivariate normal variational distribution $q_{\boldsymbol{w}}(\boldsymbol{\theta}) := \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

for $i = 0, \ldots, N-1$ ▷ `iteration loop`

do

  for $j = 0, \ldots, B-1$ ▷ `batch loop`

  do

    Sample $\boldsymbol{\theta}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$

    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^{(j)}|\boldsymbol{x})$

    $\boldsymbol{\varepsilon} \leftarrow \boldsymbol{\Sigma}_i \boldsymbol{g} - \boldsymbol{\mu}_i + \boldsymbol{\theta}$

    Solve $\rho(1+\rho) = \boldsymbol{g}^\top \boldsymbol{\Sigma}_i \boldsymbol{g} + \left[(\boldsymbol{\mu}_i - \boldsymbol{\theta})^\top \boldsymbol{g}\right]^2$ for $\rho > 0$

    $\boldsymbol{\delta\mu}^{(j)} \leftarrow \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_i - \boldsymbol{\theta})\boldsymbol{g}^\top}{1+\rho+(\boldsymbol{\mu}_i-\boldsymbol{\theta})^\top \boldsymbol{g}}\right]\boldsymbol{\varepsilon}$

    $\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} \leftarrow \boldsymbol{\mu_i} + \boldsymbol{\delta\mu}^{(j)}$

    $\boldsymbol{\delta\Sigma}^{(j)} \leftarrow (\boldsymbol{\mu}_i - \boldsymbol{\theta})(\boldsymbol{\mu}_i - \boldsymbol{\theta})^\top - (\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} - \boldsymbol{\theta})(\boldsymbol{\mu}_{\boldsymbol{i}}^{(j)} - \boldsymbol{\theta})^\top$

  end

  Update $\boldsymbol{\mu_{i+1}} \leftarrow \boldsymbol{\mu_i} + \sum_j \boldsymbol{\delta\mu^{(j)}}/B$

  Update $\boldsymbol{\Sigma_{i+1}} \leftarrow \boldsymbol{\Sigma_i} + \sum_j \boldsymbol{\delta\Sigma^{(j)}}/B$

end

$q_{\boldsymbol{w}}(\boldsymbol{\theta}) \leftarrow \mathcal{N}(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$

Theorem 2.2. (GSM-VI updates) Let $p(\boldsymbol{\theta}, \boldsymbol{x})$ be given for some $\boldsymbol{\theta} \in \mathbb{R}^d$, and let $q_{\boldsymbol{w}^t}(\boldsymbol{\theta})$ and $q_{\boldsymbol{w}}(\boldsymbol{\theta})$ be multivariate normal distributions with means $\boldsymbol{\mu}_t$ and $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}_t$ and $\boldsymbol{\Sigma}$, respectively. As shorthand, let $\boldsymbol{g}_t := \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t, \boldsymbol{x})$ and let

$$\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1} = \underset{\boldsymbol{\mu}, \boldsymbol{\Sigma} \succeq 0}{\operatorname{argmin}} \Big[\mathrm{KL}(q_t, q)\Big] \quad \text{such that} \quad \nabla_{\boldsymbol{\theta}} \log q(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t, \boldsymbol{x}). \tag{7}$$

The solution to eq. (7) is given by Eqs. 5 and 6 where

$$\boldsymbol{A}_t := \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_t - \boldsymbol{\theta}_t)\boldsymbol{g}_t^\top}{1+\rho+(\boldsymbol{\mu}_t - \boldsymbol{\theta}_t)^\top \boldsymbol{g}_t}\right]\boldsymbol{\Sigma}_t\,, \tag{8}$$

and ρ is the positive root of the quadratic equation

$$\rho(1+\rho) = \boldsymbol{g}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{g}_t + \left[(\boldsymbol{\mu}_t - \boldsymbol{\theta}_t)^\top \boldsymbol{g}_t\right]^2. \tag{9}$$

With the definition of $\boldsymbol{A}_t$ in Eq. 8 we can see that the computational complexity of updating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ via Eqs. 5 and 6 is $\mathcal{O}(d^2)$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ and we assume the cost of computing the gradients is $\mathcal{O}(d)$. Note this is the best possible iteration complexity we can hope for, since we store and maintain the full covariance matrix of d^2 elements. (The proof is in the appendix.)

Algorithm 1 presents the full GSM-VI algorithm. Here we also use mini-batching, where we average over $B \in \mathbb{N}$ independently sampled updates of Eqs. 5 and 6 before updating the mean and covariance.

3 Empirical Studies

We evaluate the performance of GSM-VI in different settings. GSM-VI uses a multivariate Gaussian distribution as its variational family. We separately investigate when the target posterior is in this family and when it is not.

Algorithmic details and comparisons. We compare GSM-VI with a reparameterization variant of BBVI as the baseline, similar to [19]. BBVI uses the same multivariate Gaussian variational family, which we fit by maximizing the ELBO. (Maximizing the ELBO is equivalent to minimizing KL). We

Algorithm 2: Black-box variational inference

Input : Initial mean estimate $\boldsymbol{\mu_0}$, Initial covariance estimate $\boldsymbol{\Sigma_0}$, target distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$, number of iterations N, batch size B, learning rate ϵ

Output : Multivariate normal variational distribution $q_{\boldsymbol{w}}(\boldsymbol{\theta}) := \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$q_{\boldsymbol{w}} \leftarrow \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$;

for $i = 0, \ldots, N-1$ ▷ `iteration loop`

do

$\{\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(B)}\} \sim q_{\boldsymbol{w}}(\theta)$ ▷ `Sample a batch of B points;`

$\text{ELBO} = \sum_j \log(p(\boldsymbol{\theta}^{(j)}, \boldsymbol{x}) - \log q_{\boldsymbol{w}}(\theta^{(j)})$;

$\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon \nabla_{\boldsymbol{w}} \text{ELBO}$ ▷ `Optimization step, we use ADAM;`

end

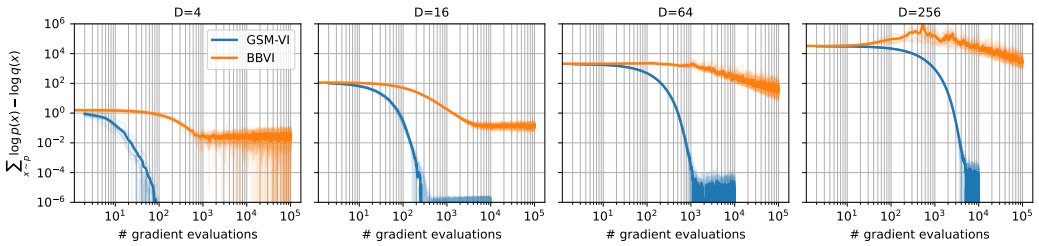


Figure 2: Scaling with dimensions: evolution of FKL with the number of gradient evaluations of the target distribution, which is a Gaussian distribution with dense covariance. Different panels show the results for different dimensions D of the distribution, specified in the title. Translucent lines show the scatter of 10 different runs and the solid line shows the average.

use the ADAM optimizer [17] with default settings but vary the learning rate between 10^{-1} and 10^{-3}. We report results only for the best performing setting. The full BBVI algorithm is in Algorithm 2.

The only free parameter in GSM-VI is the batch size B. We find that $B = 2$ is better than $B = 1$, but there is no improvement beyond that. In all studies, we report results for $B = 2$.

Both algorithms require an initial variational distribution. Unless specified otherwise, we initialize the variational distribution with zero mean and identity covariance matrix.

Evaluation metric. GSM-VI does not explicitly minimize any loss function. Hence to compare its performance against BBVI, we estimate empirical divergences between the variational and the target distribution and show their evolution with the number of gradient evaluations. In the experiments with synthetic models in Sections 3, 3.1, and 3.1 we have access to the true distribution; so we measure the forward KL divergence (FKL) empirically $\left(\text{FKL} = \sum_{\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})} \log p(\boldsymbol{\theta}_i) - \log q(\boldsymbol{\theta}_i)\right)$. To reduce stochasticity, we always use the same pre-generated set of 1000 samples from the target distribution. In Section 3.3, we do not have access to the samples from the target distribution; so we monitor the negative ELBO. In all experiments, we show the results for 10 independent runs.

3.1 GSM-VI for Gaussian approximation

We begin by studying GSM-VI where the target distribution is also a multivariate Gaussian.

Scaling with dimensions. How does GSM-VI scale with respect to the dimensions of the sample space? Figure 2 shows the convergence of FKL for GSM-VI and BBVI as the dimension (D) of the sample space increases. Empirically, we find that the number of iterations required for convergence increases almost linearly with dimensions for GSM. The scaling for BBVI is worse, and it requires 100 times more iterations even for small problems ($D < 64$), while also converging to a sub-optimal solution as measured by the FKL metric.

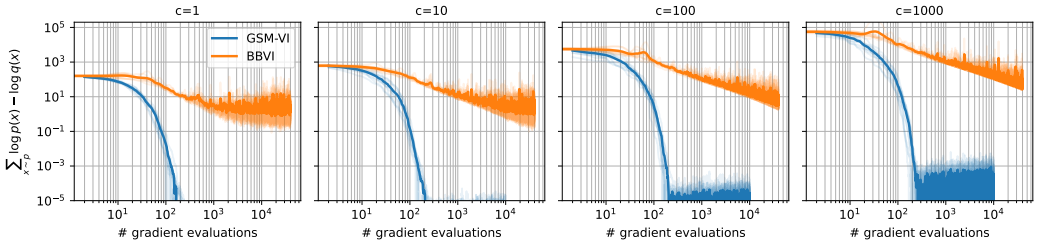


Figure 3: Impact of condition number: evolution of FKL with the number of gradient evaluations of the target distribution. The target is a 10-dimensional Gaussian albeit with a dense covariance matrix of different condition numbers c specified in the title of different panels. Translucent lines show the scatter of 10 different runs and the solid line shows the average.

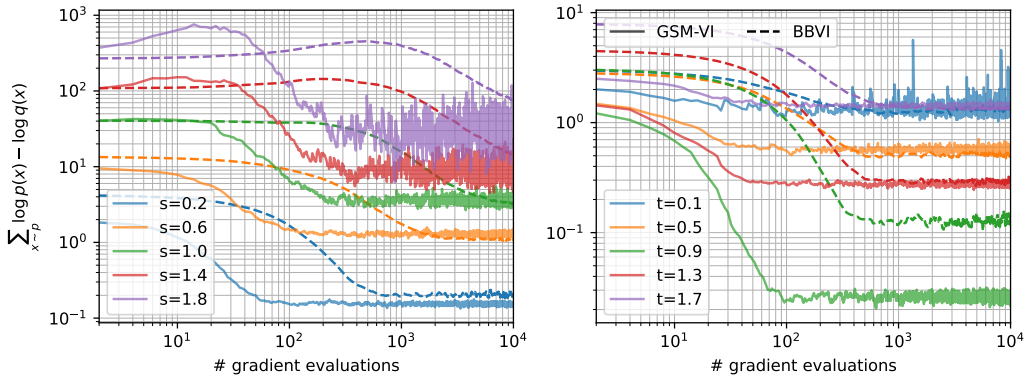


Figure 4: Impact of non-Gaussianity: evolution of FKL with the number of gradient evaluations for Sinh-arcsinh distributions with 10-dimensional dense Gaussian as the base distribution. Gaussian distribution has $s = 0$, $t = 1$. In the left panel, we vary skewness s while fixing $t = 1$, and in the right panel we vary the tail-weight t with skewness fixed to $s = 0$. Solid lines are the results for GSM, dashed for BBVI.

Impact of condition number. What is the impact of the conditioning of the target distribution? We again consider a Gaussian target distribution, but vary the condition number of its covariance matrix by fixing its smallest eigenvalue to 0.1, and scaling the largest eigenvalue to $0.1 \times c$. Figure 3 shows the results for a 10 dimensional Gaussian where we vary the condition number c from 1 to 1000. Convergence of GSM-VI seems to be largely insensitive to the condition number of the covariance matrix. BBVI on the other hand struggles with poorly conditioned problems, and it does not converge for $c > 100$ even with 100 times more iterations than GSM.

3.2 GSM-VI for non-Gaussian target distributions

GSM-VI was designed to solve the exact score-matching equations Eq. 3, which only have a solution when the family of variational distributions contains the target distribution (see Lemma 2.1). Here we investigate the sensitivity of GSM-VI to this assumption by fitting non-Gaussian target distributions with varying degrees on non-Gaussianity. Specifically, we suppose that the target has a multivariate `Sinh-arcsinh` normal distribution [15]

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}); \quad \boldsymbol{x} = \sinh\left(\frac{1}{t}\left[\sinh^{-1}(\boldsymbol{z}) + s\right]\right) \tag{10}$$

where the scalar parameters s and t control, respectively, the skewness and the heaviness of the tails, and the choices $s = 0$ and $t = 1$ reduce a Gaussian distribution as a special case.

Figure 4 shows the result for fitting the variational Gaussian to a 10-dimensional Sinh-arcsinh normal distribution for different values of s and t. As the target departs further from Gaussianity, the quality

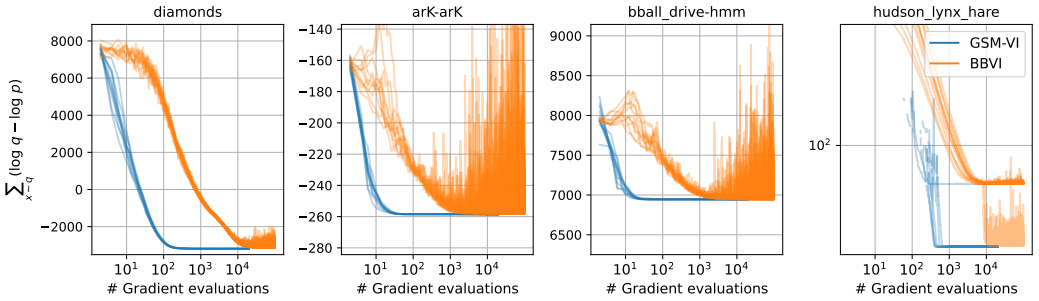


Figure 5: Models from `posteriordb`: Convergence of the ELBO for four models with multivariate normal posteriors. We show results for 10 runs.

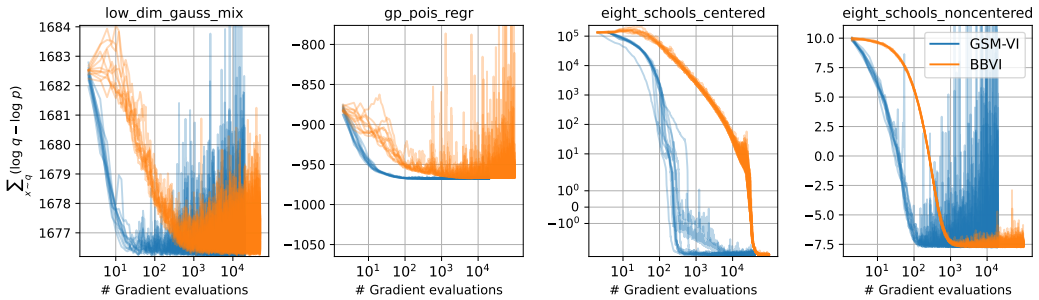


Figure 6: Models from `posteriordb`: Covergence of ELBO for four models with non-Gaussian posteriors. We show results for 10 runs.

of variational fit worsens for both GSM-VI (solid lines) and BBVI (dashed lines), but they converge to a fit of similar quality in terms of average FKL. GSM converges to this solution at least 10 times faster than BBVI. For highly non-Gaussian targets ($s \geq 1$ or $|t-1| \geq 0.8$), we have found that GSM-VI does not converge to a fixed point, and it can experience oscillations that are larger in amplitude than BBVI, see for instance $s = 1.8$ and $t = 0.1$ on the left and right of Figure 4, respectively.

3.3 GSM-VI on real-world data

We evaluate GSM-VI for approximate on real-world data with 8 models from the `posteriordb` database [21]. The database provides the `Stan` code, data and reference posterior samples, and we use `bridgestan` to access the gradients of these models [5, 26]. We study the following models: `diamonds` (generalized linear models), `hudson-lynx-hare` (differential equation dynamics), `bball-drive` (hidden Markov models) and `arK` (time-series), `eight-schools-centered` and `non-centered` (hierarchical meta-analysis), `gp-pois-regr` (Gaussian processes), `low-dim-gauss-mix` (Gaussian mixture).

For each model (except `hudson-lynx-hare`), we initialize the variational parameter $\boldsymbol{\mu}_0$ at the mode of the distribution, and we set $\boldsymbol{\Sigma}_0 = 0.1\, \boldsymbol{I}_d$ where $\boldsymbol{I}_d$ is the identity matrix of dimension d. For `hudson-lynx-hare`, we initialize the variational distribution as standard normal. We also experimented with other initializations. We find that they do not qualitatively change the conclusions, but can have larger variance between different runs.

We show the evolution of the ELBO for 10 runs of these models. Four of the models have posteriors that can be fit with multivariate normal distribution: `diamonds`, `hudson-lynx-hare`, `bball-drive`, and `arK`. Figure 5 shows the result for these models. The other models have non-Gaussian posteriors: `eight-schools-centered`, `eight-schools-non-centered`, `gp-pois-regr`,, and `low-dim-gauss-mix`. Figure 6 shows the results.

Overall, GSM-VI outperforms BBVI by a factor of 10-100x. When the target posterior is Gaussian, GSM-VI leads to more stable solutions. When the target is non-Gaussian, it converges to the same quality of variational approximation as BBVI. Further, though the ELBO estimate is noisy at the convergence, the 1-D marginals and moments of parameters remain stable.

4 Conclusion and Future Work

In this paper we proposed Gaussian score matching VI (GSM-VI), a new approach for VI when the variational family is multivariate Gaussian. GSM-VI is not based on minimizing a divergence or loss function between the variational and target distribution; instead, it repeatedly solves the exact score matching equations with closed-form updates for the mean and covariance matrix of the variational distribution.

Unlike approaches that are rooted in stochastic gradient descent, GSM-VI does not require the tuning of step-size hyper-parameters. It has only one free parameter, the batch size, and we found a batch-size of 2 to perform competitively across all experiments. Another choice is how to initialize the variational distribution. For the experiments in this paper, we initialized the covariance matrix as the identity matrix, but additional gains could potentially be made with more informed choices derived from a Laplace approximation or L-BFGS Hessian approximation [31].

We evaluated the performance of GSM-VI on synthetic targets and real-world models from `posteriordb`. In general, we found that it requires 10-100x fewer gradient evaluations than BBVI for the target distribution to converge. When the target distribution is itself multivariate Gaussian, we observed that GSM-VI scales almost *linearly* with dimensionality, which is significantly better than BBVI, and that GSM-VI is almost insensitive to the condition number of the target covariance matrix. Compared to BBVI, we also found that GSM-VI converges more quickly to a solution with a larger ELBO, which is surprising given that BBVI explicitly maximizes the ELBO.

GSM-VI is derived from a principled goal and justification, and the empirical studies indicate that it is a promising method. An important avenue for future work is to provide a proof that GSM-VI converges. We note that good convergence results have been obtained for analogous methods that project onto interpolation equations for empirical risk minimization. For instance the Stochastic Polyak Step achieves the min-max optimal rates of convergence for SGD [20]. Note that convergence of VI is a generally challenging problem, with no known rates of convergence even for BBVI [9, 10].

In another avenue of future work, the score-matching VI idea can potentially be used to design other methods for VI. As one example, we can consider non-Gaussian variational approximations, such as those in the exponential family. As another example, if the variational family is a mixture of Gaussians, we can employ GSM-VI to update the individual components of the mixture.

References

[1] A. Agrawal, D. R. Sheldon, and J. Domke. Advances in black-box vi: normalizing flows, importance weighting, and optimization. *Neural Information Processing Systems*, 2020.

[2] L. Berrada, A. Zisserman, and M. P. Kumar. Training neural networks for and by interpolation. In *Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

[3] E. Bingham, J. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. Goodman. Pyro: Deep universal probabilistic programming. *arXiv:1810.09538*, 2018.

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

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

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[7] A. K. Dhaka, A. Catalina, M. Welandawe, M. Riis Andersen, J. Huggins, and A. Vehtari. Challenges and Opportunities in High-dimensional Variational Inference. art. arXiv:2103.01085, March 2021.

[8] A. Dieng, D. Tran, R. Ranganath, J. Paisley, and D. Blei. Variational inference via χ upper bound minimization. In *Neural Information Processing Systems*, 2017.

[9] J. Domke. Provable gradient variance guarantees for black-box variational inference. In *Neural Information Processing Systems*, volume 32, 2019.

[10] J. Domke. Provable smoothness guarantees for black-box variational inference. In *International Conference on Machine Learning*, volume 119, pages 2587–2596. PMLR, 2020.

[11] M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *International Conference on Machine Learning,*, volume 307, pages 264–271, 2008.

[12] R. M. Gower, A. Defazio, and M. Rabbat. Stochastic polyak stepsize with a moving target. *arXiv:2106.11851*, 2021.

[13] M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[14] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[15] C. Jones and A. Pewsey. The Sinh-Arcsinh Normal Distribution. *Significance*, 16(2):6–7, 04 2019. ISSN 1740-9705.

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

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[18] J. Knoblauch, J. Jewson, and T. Damoulas. An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23: 1–109, 2022.

[19] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic Differentiation Variational Inference. art. arXiv:1603.00788, March 2016.

[20] N. Loizou, S. Vaswani, I. Laradji, and S. Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. *International Conference on Artificial Intelligence and Statistics*, 2021.

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

[22] T. Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence (UAI)*, pages 362–369, 2001.

[23] C. Naesseth, F. Lindsten, and D. Blei. Markovian score climbing: Variational inference with KL(p || q). In *Neural Information Processing Systems*, 2020.

[24] R. Ranganath, S. Gerrish, and D. M. Blei. Black Box Variational Inference. art. arXiv:1401.0118, December 2013.

[25] R. Ranganath, J. Altosaar, D. Tran, and D. Blei. Operator variational inference. In *Neural Information Processing Systems*. 2016.

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

[27] J. Salvatier, Wiecki T., and C. Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2016.

[28] U. Seljak and B. Yu. Posterior inference unchained with EL_2O. art. arXiv:1901.04454, January 2019.

[29] M. W., M. R. Andersen, A. Vehtari, and J. H. Huggins. Robust, automated, and accurate black-box variational inference. *arXiv:2203.15945*, 2022.

[30] Y. Yang, R. Martin, and H. Bondell. Variational approximations using Fisher divergence. *arXiv e-prints*, art. arXiv:1905.05284, May 2019.

[31] L. Zhang, B. Carpenter, A. Gelman, and A. Vehtari. Pathfinder: Parallel quasi-newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49, 2022.

A Proof of Lemma 2.1

Lemma 2.1. The parameter $\boldsymbol{w}^*$ satisfies

$$\nabla_{\boldsymbol{\theta}} \log\, q_{\boldsymbol{w}^*}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log\, p(\boldsymbol{\theta}, \boldsymbol{x}), \qquad \forall \boldsymbol{\theta}, \tag{3}$$

if and only if $\boldsymbol{w}^*$ also satisfies Eq. 1.

Proof. (1) $\implies$ (3): Differentiating both sides of (1) in $\boldsymbol{\theta}$ gives

$$\begin{aligned}\nabla_\theta \log q_{\boldsymbol{w}}(\boldsymbol{\theta}) &= \nabla_\theta \log p(\boldsymbol{\theta}|\boldsymbol{x}) = \nabla_\theta \left(\log p(\boldsymbol{\theta}, \boldsymbol{x}) - \log p(\boldsymbol{x})\right) \\ &= \nabla_\theta \log p(\boldsymbol{\theta}, \boldsymbol{x}), \qquad \forall \boldsymbol{\theta}.\end{aligned}$$

(3) $\implies$ (1): The inverse implication follows by using that $\nabla_\theta \log p(\boldsymbol{\theta}|\boldsymbol{x}) = \nabla_\theta \log p(\boldsymbol{\theta}, \boldsymbol{x})$, as we did in the above, and then integrating both sides of (3) in θ, which gives

$$\log q_{\boldsymbol{w}}(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}|\boldsymbol{x}) + C, \qquad \forall \boldsymbol{\theta},$$

where C is some unknown constant. By exponentiating both sides and integrating in $\boldsymbol{\theta}$ we have that

$$1 = \int_{\boldsymbol{\theta}} q_{\boldsymbol{w}}(\boldsymbol{\theta}) d\boldsymbol{\theta} = e^C \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{x}) d\boldsymbol{\theta} = e^C.$$

Consequently $C = 0$, which gives our result. □

B Proof of Theorem 2.2

Here we give the proof for Theorem 2.2. We also re-introduce the theorem with a simplified notation, where we use $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ to denote the mean and covariance at the previous time step of the method, thus dropping the iteration counter t.

Theorem B.1. (GSM updates) Let $p(\boldsymbol{\theta}, \boldsymbol{x})$ be given for some $\boldsymbol{\theta} \in \mathbb{R}^d$, and let $q_0(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ be the multivariate normal distributions, respectively, with means $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}$. We seek the distribution

$$\underset{\boldsymbol{\mu}, \boldsymbol{\Sigma}=\boldsymbol{\Sigma}^\top}{\arg\min} \Big[\mathrm{KL}(q_0, q)\Big] \quad \text{such that} \quad \nabla_{\boldsymbol{\theta}} \log q(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \boldsymbol{x}). \tag{11}$$

As shorthand, let $\boldsymbol{g} := \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \boldsymbol{x})$, and let ρ be the positive root of the quadratic equation

$$\rho(1+\rho) \;=\; \boldsymbol{g}^\top \boldsymbol{\Sigma}_0 \boldsymbol{g} + \left[(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top \boldsymbol{g}\right]^2. \tag{12}$$

Then the solution is given by the following closed-form updates:

$$\boldsymbol{\mu} \;=\; \boldsymbol{\mu}_0 + \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top}{1 + \rho + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top \boldsymbol{g}}\right] \boldsymbol{\Sigma}_0 \left(\boldsymbol{g} - \nabla_{\boldsymbol{\theta}} \log q_0(\boldsymbol{\theta}_{\mathbf{0}})\right), \tag{13}$$

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top - (\boldsymbol{\mu} - \boldsymbol{\theta})(\boldsymbol{\mu} - \boldsymbol{\theta})^\top. \tag{14}$$

Furthermore, if $\boldsymbol{\Sigma}_0$ is symmetric positive definite then so is $\boldsymbol{\Sigma}$.

Proof. The constraint in this optimization is given by

$$\begin{aligned}\boldsymbol{g} &= \nabla_{\boldsymbol{\theta}} \log q(\boldsymbol{\theta}) && (15)\\ &= \nabla_{\boldsymbol{\theta}} \left[-\tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}) - \tfrac{1}{2}\log\left((2\pi)^d |\boldsymbol{\Sigma}|\right)\right] && (16)\\ &= -\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}). && (17)\end{aligned}$$

The KL divergence is given by

$$\mathrm{KL}(q_0, q) \;=\; \frac{1}{2}\left\{\mathrm{tr}[\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_0] + \log\frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_0|} + (\boldsymbol{\mu} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) - d\right\}. \tag{18}$$

Dropping irrelevant terms from the optimization, we obtain the Lagrangian

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\lambda}) \;=\; \frac{1}{2}\left\{\mathrm{tr}[\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_0] - \log|\boldsymbol{\Sigma}^{-1}| + (\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\right\} + \boldsymbol{\lambda}^\top\big(\boldsymbol{g} - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\theta})\big). \tag{19}$$

It is easier to optimize the matrix $\boldsymbol{\Sigma}^{-1}$ instead of $\boldsymbol{\Sigma}$. We can enforce the symmetry of $\boldsymbol{\Sigma}^{-1}$ by writing

$$\boldsymbol{\Sigma^{-1}} \;=\; \tfrac{1}{2}\left(\boldsymbol{\Phi} + \boldsymbol{\Phi}^\top\right) \tag{20}$$

and performing an unconstrained optimization over $\boldsymbol{\Phi}$. With respect to the latter, the gradients of the Lagrangian are given by

$$\frac{\partial\mathcal{L}}{\partial\Phi_{ij}} \;=\; \sum_{kl}\left(\frac{\partial\mathcal{L}}{\partial\Sigma^{-1}_{kl}}\right)\left(\frac{\partial\Sigma^{-1}_{kl}}{\partial\Phi_{ij}}\right) \;=\; \sum_{kl}\left(\frac{\partial\mathcal{L}}{\partial\Sigma^{-1}_{kl}}\right)\left(\frac{1}{2}\delta_{ki}\delta_{lj} + \frac{1}{2}\delta_{kj}\delta_{li}\right) \;=\; \frac{1}{2}\left(\frac{\partial\mathcal{L}}{\partial\Sigma^{-1}_{ij}} + \frac{\partial\mathcal{L}}{\partial\Sigma^{-1}_{ji}}\right). \tag{21}$$

Next we examine where the gradients of the Lagrangian vanish:

$$\boldsymbol{0} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}} \;\;\Longrightarrow\;\; \boldsymbol{0} \;=\; \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu_0}) \;-\; \boldsymbol{\Sigma}^{-1}\boldsymbol{\lambda} \;\;\Longrightarrow\;\; \boxed{\boldsymbol{\lambda} = \boldsymbol{\mu}-\boldsymbol{\mu}_0} \tag{22}$$

$$\boldsymbol{0} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{\lambda}} \;\;\Longrightarrow\;\; \boldsymbol{0} \;=\; \boldsymbol{g} - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\theta}) \;\;\Longrightarrow\;\; \boxed{\boldsymbol{\mu}-\boldsymbol{\theta} = \boldsymbol{\Sigma g}} \tag{23}$$

$$\boldsymbol{0} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{\Phi}} \;\;\Longrightarrow\;\; \boldsymbol{0} \;=\; \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma} - (\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top - \left[\boldsymbol{\lambda}(\boldsymbol{\mu}-\boldsymbol{\theta})^\top + (\boldsymbol{\mu}-\boldsymbol{\theta})\boldsymbol{\lambda}^\top\right], \tag{24}$$

$$\Longrightarrow\;\; \boxed{\boldsymbol{\Sigma} \;=\; \boldsymbol{\Sigma}_0 \;+\; (\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top \;-\; \boldsymbol{\lambda}(\boldsymbol{\mu}-\boldsymbol{\theta})^\top \;-\; (\boldsymbol{\mu}-\boldsymbol{\theta})\boldsymbol{\lambda}^\top} \tag{25}$$

We claim that these equations (though nonlinear) can be solved in closed form. The first step is to eliminate $\boldsymbol{\lambda}$ from eq. (25) using eq. (22). In this way we find

$$\begin{aligned}
\boldsymbol{\Sigma} \;&=\; \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top - (\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\theta})^\top - (\boldsymbol{\mu}-\boldsymbol{\theta})(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^\top && (26)\\
&=\; \boldsymbol{\Sigma}_0 - \boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\mu}\boldsymbol{\theta}^\top + \boldsymbol{\theta}\boldsymbol{\mu}^\top + \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\top - \boldsymbol{\mu}_0\boldsymbol{\theta}^\top - \boldsymbol{\theta}\boldsymbol{\mu}_0^\top && (27)\\
&=\; \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0-\boldsymbol{\theta})(\boldsymbol{\mu}_0-\boldsymbol{\theta})^\top - (\boldsymbol{\mu}-\boldsymbol{\theta})(\boldsymbol{\mu}-\boldsymbol{\theta})^\top. && (28)
\end{aligned}$$

It is worth highlighting the form of this equation:

$$\boldsymbol{\Sigma} \;=\; \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0-\boldsymbol{\theta})(\boldsymbol{\mu}_0-\boldsymbol{\theta})^\top - (\boldsymbol{\mu}-\boldsymbol{\theta})(\boldsymbol{\mu}-\boldsymbol{\theta})^\top$$

This is a simple rank-two update for $\boldsymbol{\Sigma}$. Note that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma_0}$ if $\boldsymbol{\mu} = \boldsymbol{\mu}_0$; also, the solution for $\boldsymbol{\Sigma}$ is determined by the solution for $\boldsymbol{\mu}$.

Ultimately we must solve for $\boldsymbol{\mu}$, but first it is useful to solve for the intermediate quantity $\boldsymbol{g}^\top\boldsymbol{\Sigma g} > 0$. From eq. (28), we obtain

$$\boldsymbol{g}^\top\boldsymbol{\Sigma g} \;=\; \boldsymbol{g}^\top\boldsymbol{\Sigma}_0\boldsymbol{g} + \left[(\boldsymbol{\mu}_0-\boldsymbol{\theta})^\top\boldsymbol{g}\right]^2 - \left[(\boldsymbol{\mu}-\boldsymbol{\theta})^\top\boldsymbol{g}\right]^2, \tag{29}$$

and from eq. (23), we obtain

$$\boldsymbol{g}^\top\boldsymbol{\Sigma g} \;=\; \boldsymbol{g}^\top\boldsymbol{\Sigma}_0\boldsymbol{g} + \left[(\boldsymbol{\mu}_0-\boldsymbol{\theta})^\top\boldsymbol{g}\right]^2 - \left(\boldsymbol{g}^\top\boldsymbol{\Sigma g}\right)^2. \tag{30}$$

As shorthand, let $\rho = \boldsymbol{g}^\top\boldsymbol{\Sigma g}$. Then from eq. (30) we see that ρ satisfies the quadratic equation

$$\rho(1+\rho) \;=\; \boldsymbol{g}^\top\boldsymbol{\Sigma}_0\boldsymbol{g} + \left[(\boldsymbol{\mu}_0-\boldsymbol{\theta})^\top\boldsymbol{g}\right]^2.$$

Note that there are no unknowns on the right side of this equation. The correct solution is given by the positive root since $\rho = \boldsymbol{g}^\top\boldsymbol{\Sigma g} > 0$. Also note that $\rho = (\boldsymbol{\mu}-\boldsymbol{\theta})^\top\boldsymbol{g}$ from eq. (23).

It is useful to define one final intermediate quantity before solving for $\boldsymbol{\mu}$. Let

$$\boldsymbol{\varepsilon}_0 \;=\; \boldsymbol{\Sigma}_0\boldsymbol{g} - \boldsymbol{\mu}_0 + \boldsymbol{\theta}.$$

Note that $\boldsymbol{\varepsilon}_0$ simply measures the degree to which the parameters of $q_0(\boldsymbol{\theta})$ violate the desired constraint $\nabla_{\boldsymbol{w}}\log q(\boldsymbol{\theta}) = \nabla_{\boldsymbol{w}}\log p(\boldsymbol{\theta}, \boldsymbol{y})$. Put another way, if $\boldsymbol{\varepsilon}_0 = \boldsymbol{0}$, then we have the trivial solution $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0$.

Now we have everything to express the solution for $\boldsymbol{\mu}$ in a highly intuitive form; in particular, it will be immediately evident that $\boldsymbol{\mu} \to \boldsymbol{\mu}_0$ as $\boldsymbol{\varepsilon}_0 \to \mathbf{0}$. Starting from eqs. (23) and (28), we find

$$\begin{aligned}
\boldsymbol{\mu} - \boldsymbol{\mu}_0 &= \boldsymbol{\theta} - \boldsymbol{\mu}_0 + \boldsymbol{\Sigma}\boldsymbol{g}, && (31)\\
&= \boldsymbol{\theta} - \boldsymbol{\mu}_0 + \left[\boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top - (\boldsymbol{\mu} - \boldsymbol{\theta})(\boldsymbol{\mu} - \boldsymbol{\theta})^\top\right]\boldsymbol{g}, && (32)\\
&= \boldsymbol{\varepsilon}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g} - (\boldsymbol{\mu} - \boldsymbol{\theta})(\boldsymbol{\mu} - \boldsymbol{\theta})^\top\boldsymbol{g}, && (33)\\
&= \boldsymbol{\varepsilon}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g} - (\boldsymbol{\mu} - \boldsymbol{\mu}_0 + \boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu} - \boldsymbol{\theta})^\top\boldsymbol{g}, && (34)\\
&= \boldsymbol{\varepsilon}_0 - \rho(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})[(\boldsymbol{\mu}_0 - \boldsymbol{\theta}) - (\boldsymbol{\mu} - \boldsymbol{\theta})^\top]\boldsymbol{g}, && (35)\\
&= \boldsymbol{\varepsilon}_0 - \rho(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\mu})^\top\boldsymbol{g}, && (36)\\
&= \boldsymbol{\varepsilon}_0 - (\rho\mathbf{I} + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}). && (37)
\end{aligned}$$

Note what has happened here: eq. (32) is a system of *nonlinear* equations for $\boldsymbol{\mu}$, but in eq. (37), all the nonlinearity has been expressed in terms of ρ. Since ρ can be determined via eq. (30), we arrive effectively at a system of *linear* equations for $\boldsymbol{\mu}$. Collecting terms, we obtain

$$\left[(1+\rho)\mathbf{I} + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top\right](\boldsymbol{\mu} - \boldsymbol{\mu}_0) = \boldsymbol{\varepsilon}_0. \tag{38}$$

We thus arrive at the closed-form update

$$\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \left[(1+\rho)\mathbf{I} + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top\right]^{-1}\boldsymbol{\varepsilon}_\mathbf{0} \tag{39}$$

It is evident from this update that $\boldsymbol{\mu} \to \boldsymbol{\mu}_0$ as $\boldsymbol{\varepsilon}_0 \to \mathbf{0}$. The matrix inverse in this update can also be computed efficiently from the Woodbury matrix identity.

In sum, the joint update for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be efficiently computed as follows:

1. Set $\boldsymbol{g} = \nabla_{\boldsymbol{w}} \log p(\boldsymbol{\theta}, \boldsymbol{y})$ and $\boldsymbol{\varepsilon}_0 = \boldsymbol{\Sigma}_0\boldsymbol{g} - \boldsymbol{\mu}_0 + \boldsymbol{\theta}$.
2. Solve $\rho(1+\rho) = \boldsymbol{g}^\top\boldsymbol{\Sigma}_0\boldsymbol{g} + \left[(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g}\right]^2$ for $\rho > 0$.
3. Compute $\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \left[(1+\rho)\mathbf{I} + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top\right]^{-1}\boldsymbol{\varepsilon}_\mathbf{0}$.
4. Compute $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top - (\boldsymbol{\mu} - \boldsymbol{\theta})(\boldsymbol{\mu} - \boldsymbol{\theta})^\top$.

Solving the quadratic in 2. for ρ we have the positive

$$\boxed{\rho = \frac{\sqrt{1 + 4(\boldsymbol{g}^\top\boldsymbol{\Sigma}_0\boldsymbol{g} + \left[(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g}\right]^2)} - 1}{2}}$$

Solving the above linear equation for μ and using the Sherman Morrison formula

$$\left[a\mathbf{I} + \boldsymbol{u}\boldsymbol{g}^\top\right]^{-1} = \frac{1}{a}\left(\mathbf{I} - \frac{\boldsymbol{u}\boldsymbol{g}^\top}{a + \boldsymbol{u}^\top\boldsymbol{g}}\right), \quad \text{for every } \boldsymbol{u}, \boldsymbol{g}, a$$

gives

$$\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top}{1 + \rho + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g}}\right]\boldsymbol{\varepsilon}_\mathbf{0}. \tag{40}$$

Using that $\nabla_{\boldsymbol{\theta}} \log q_0(\boldsymbol{\theta}_\mathbf{0}) = -\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_0)$ we have that

$$\boldsymbol{\varepsilon}_\mathbf{0} = \boldsymbol{\Sigma}_0\left(\boldsymbol{g} - \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu}_0 + \boldsymbol{\theta})\right) = \boldsymbol{\Sigma}_0\left(\boldsymbol{g} - \nabla_{\boldsymbol{\theta}} \log q_0(\boldsymbol{\theta}_\mathbf{0})\right).$$

Finally substituting out $\boldsymbol{\varepsilon}_\mathbf{0}$ in (40) the result

$$\boxed{\boldsymbol{\mu} = \boldsymbol{\mu}_0 + \frac{1}{1+\rho}\left[\mathbf{I} - \frac{(\boldsymbol{\mu}_0 - \boldsymbol{\theta})\boldsymbol{g}^\top}{1 + \rho + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top\boldsymbol{g}}\right]\boldsymbol{\Sigma}_0\left(\boldsymbol{g} - \nabla_{\boldsymbol{\theta}} \log q_0(\boldsymbol{\theta}_\mathbf{0})\right).}$$

Proof that $\boldsymbol{\Sigma_0}$ $p.s.d \implies \boldsymbol{\Sigma}$ $p.s.d$. It remains to prove that our solution for $\boldsymbol{\Sigma}$ is positive-definite, or equivalently, that all of its eigenvalues are positive. We begin by rewriting our results for $\boldsymbol{\Sigma}$ in eq. (28) and ρ in eq. (30) in a more convenient form. As shorthand, let

$$\mathbf{M}_0 \;=\; \boldsymbol{\Sigma}_0 + (\boldsymbol{\mu}_0 - \boldsymbol{\theta})(\boldsymbol{\mu}_0 - \boldsymbol{\theta})^\top, \tag{41}$$

so that $\mathbf{M}_0$ captures the first two terms on the right side of eq. (28). Note that $\mathbf{M}_0$ is positive-definite, a fact that we will exploit repeatedly in what follows. In addition, recall that $\boldsymbol{\mu} - \boldsymbol{\theta} = \boldsymbol{\Sigma}\boldsymbol{g}$ from eq. (23). Thus with this notation we can rewrite eqs. (28) and (30) as

$$\boldsymbol{\Sigma} \;=\; \mathbf{M}_0 - (\boldsymbol{\Sigma}\boldsymbol{g})(\boldsymbol{\Sigma}\boldsymbol{g})^\top, \tag{42}$$

$$\rho(1+\rho) \;=\; \boldsymbol{g}^\top \mathbf{M}_0 \boldsymbol{g}. \tag{43}$$

Now let $\boldsymbol{e}$ be any normalized eigenvector of $\boldsymbol{\Sigma}$; we want to show that its corresponding eigenvalue λ_e is positive. From eq. (42), it follows that

$$\lambda_e \;=\; \boldsymbol{e}^\top \boldsymbol{\Sigma} \boldsymbol{e} \tag{44}$$

$$\;=\; \boldsymbol{e}^\top \left[\mathbf{M}_0 - (\Sigma\boldsymbol{g})(\boldsymbol{\Sigma}\boldsymbol{g})^\top\right] \boldsymbol{e} \tag{45}$$

$$\;=\; \boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{e} - \lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2. \tag{46}$$

Note that if $\boldsymbol{e}^\top \boldsymbol{g} = 0$, then it follows trivially that $\lambda_e = \boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{e} > 0$. So we only need to consider the non-trivial case $\boldsymbol{e}^\top \boldsymbol{g} \neq 0$. To proceed, we note that

$$(\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{g})^2 \;=\; (\boldsymbol{e}^\top \mathbf{M}_0^{\frac{1}{2}} \mathbf{M}_0^{\frac{1}{2}} \boldsymbol{g})^2 \;\leq\; (\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{e})(\boldsymbol{g}^\top \mathbf{M}_0 \boldsymbol{g}), \tag{47}$$

where we have used the Cauchy-Schwartz inequality to bound $(\boldsymbol{e}^\top \mathbf{M}_o \boldsymbol{g})$ in terms of $(\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{e})$, the latter of which appears in eq. (46). Substituting this inequality into eq. (46), we find that

$$\lambda_e \;\geq\; \frac{(\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{g})^2}{\boldsymbol{g}^\top \mathbf{M}_0 \boldsymbol{g}} \;-\; \lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2. \tag{48}$$

To prove that $\lambda_e > 0$ we need one more intermediate result. Focusing on the rightmost term in this equality, we note that

$$\lambda_e (\boldsymbol{e}^\top \boldsymbol{g}) \;=\; \boldsymbol{e}^\top \Sigma \boldsymbol{g} \;=\; \boldsymbol{e}^\top \left[\mathbf{M}_0 - (\boldsymbol{\Sigma}\boldsymbol{g})(\boldsymbol{\Sigma}\boldsymbol{g})^\top\right] \boldsymbol{g} \;=\; \boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{g} - \lambda_e (\boldsymbol{e}^\top \boldsymbol{g})(\boldsymbol{g}^\top \boldsymbol{\Sigma} \boldsymbol{g}), \tag{49}$$

and rearranging the terms in this equation, we find

$$\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{g} \;=\; \lambda_e (\boldsymbol{e}^\top \boldsymbol{g})(1 + \boldsymbol{g}^\top \boldsymbol{\Sigma} \boldsymbol{g}). \tag{50}$$

This intermediate result is useful because it relates the two terms on the right side of eq. (48). In particular, using eq. (50) to eliminate the term $\boldsymbol{e}^\top \mathbf{M}_0 \boldsymbol{g}$ in eq. (48), we find:

$$\begin{aligned}
\lambda_e \;&\geq\; \frac{\left[\lambda_e (\boldsymbol{e}^\top \boldsymbol{g})(1 + \boldsymbol{g}^\top \boldsymbol{\Sigma} \boldsymbol{g})\right]^2}{\boldsymbol{g}^\top \mathbf{M}_0 \boldsymbol{g}} \;-\; \lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2 \\
&=\; \lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2 \left[\frac{(1 + \boldsymbol{g}^\top \boldsymbol{\Sigma} \boldsymbol{g})^2}{\boldsymbol{g}^\top \mathbf{M}_0 \boldsymbol{g}} \;-\; 1\right] \\
&=\; \lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2 \left[\frac{(1+\rho)^2}{\rho(1+\rho)} \;-\; 1\right] \\
&=\; \frac{\lambda_e^2 (\boldsymbol{e}^\top \boldsymbol{g})^2}{\rho}, \\
&>\; 0,
\end{aligned}$$

where the final inequality follows because the individual terms λ_e^2, $(\boldsymbol{e}^\top \boldsymbol{g})^2$, and ρ are all strictly positive; note that λ_e cannot be equal to zero because this contradicts the equality in eq. (46). This completes the proof. Perhaps it is useful that this derivation also gives upper bounds on λ_e, namely

$$\frac{1}{\lambda_e} \;\geq\; \frac{(\boldsymbol{e}^\top \boldsymbol{g})^2}{\rho} \quad\implies\quad \lambda_e \;\leq\; \frac{\rho}{(\boldsymbol{e}^\top \boldsymbol{g})^2} \;=\; \frac{\boldsymbol{g}^\top \boldsymbol{\Sigma} \boldsymbol{g}}{(\boldsymbol{e}^\top \boldsymbol{g})^2}. \tag{51}$$

□