# A FIRST-ORDER METHOD FOR ESTIMATING NATURAL GRADIENTS FOR VARIATIONAL INFERENCE WITH GAUSSIANS AND GAUSSIAN MIXTURE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Variational inference with full-covariance Gaussian approximations is an important line of research, as such Gaussian variational approximations (GVAs) allow for tractable approximate inference while yielding superior approximations compared to mean-field methods. Moreover, it was recently shown, that the problem of variational inference with Gaussian mixture models can be reduced to Gaussian variational inference using VIPS, which is a procedure similar to expectation maximization. Effective approaches for Gaussian variational inference are MORE, VOGN, and VON, which are zero-order, first-order, and second-order, respectively. We focus on the first-order setting, which is arguably the most relevant for variational inference, and show that the biases added by the generalized Gauß-Newton approximation, which is applied by VOGN, can seriously affect the quality of the learned approximation. Hence, we propose gradientMORE, a method that is similar to MORE but differs by incorporating gradient information. GradientMORE achieves unbiased high-quality approximations of the Hessian that are similar to VON which has direct access to the Hessian. Our algorithm converges even in settings where VOGN does not converge. Compared to MORE, the additional information improves sample efficiency by about an order of magnitude. Furthermore, we evaluate the different approaches in the GMM setting by modifying VIPS, which has previously only been tested in combination with MORE, and show that the results from the GVA setting are transferable to GMMs, setting a new standard for GMM-based variational inference.

## 1 INTRODUCTION

A reoccurring challenge in machine learning relates to inference from intractable distributions. For example, in Bayesian inference, the intractable distribution corresponds to the posterior

$$p(\mathbf{x}|\mathcal{D}) = \frac{p(\mathbf{x})p(\mathcal{D}|\mathbf{x})}{p(\mathcal{D})}.$$

Typically, the prior $p(\mathbf{x})$ and the likelihood $p(\mathcal{D}|\mathbf{x})$ can be evaluated and differentiated, but the evidence $p(\mathcal{D})$, which normalizes the posterior, can not. Variational inference (VI) replaces the posterior with a tractable model $q(\mathbf{x})$ that minimizes the Kullback-Leibler divergence (KL),

$$D_{\mathrm{KL}}(q(\mathbf{x})||p(\mathbf{x}|\mathcal{D})) = \int_{\boldsymbol{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathcal{D})} d\boldsymbol{x}.$$

This minimization can be performed despite the intractability of $p(\mathbf{x}|\mathcal{D})$ because the normalizer $p(\mathcal{D})$ enters the objective as a constant offset that can be ignored.

Traditionally, the model family of the variational distribution $q(\mathbf{x})$ was chosen specific to the target distribution $p(\mathbf{x}|\mathcal{D})$, such that the KL could be minimized in closed form (Saul et al., 1996). However, finding appropriate model families is cumbersome and the approximations are often crude by relying on the mean-field assumption. Hence, more expressive models that can not be found in closed form, but that are independent of the target distribution, and allow for more accurate approximations have become popular (Arridge et al., 2018; Ranganath et al., 2014). Gaussian variational approximations (GVAs) with full covariance matrix are particular interesting because they

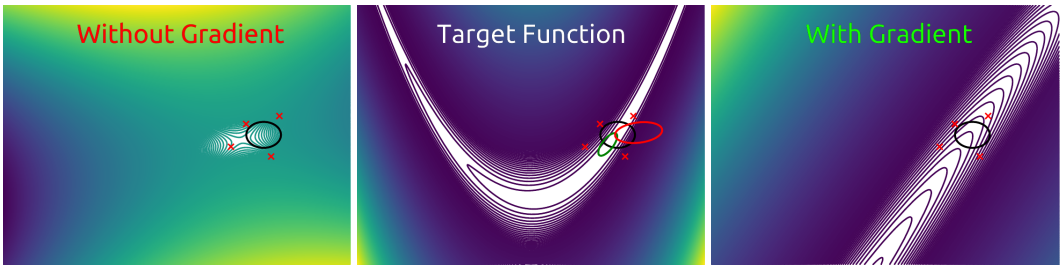



Figure 1: MORE approximates a function (middle) with a quadratic model locally to a given Gaussian search distribution (in this case isotropic, indicated by a black circle), using regression based on samples from this distribution (red markers). The left plot shows an approximation that not consider the gradients at the sample locations. Our method learns an approximation (right) that also respects the gradients at these locations. Trust-region updates of the search distribution, based on the respective surrogates, are visualized as red and green ellipses. Here, the green update, resulting from our surrogate, moves towards the optimum and better adapts the covariance towards the local curvature.

are highly tractable, for example regarding sampling, density and entropy evaluation, and marginalization. Furthermore, Arenz et al. (2020) showed, that variational inference with Gaussian mixture models (GMMs) can be reduced to the problem of learning GVAs, and, hence, optimizing GVAs can be an important building block for learning highly accurate variational approximations.

Arenz et al. (2020) adapted the stochastic search method MORE (Abdolmaleki et al., 2015) to perform this component-wise optimization. However, MORE (Abdolmaleki et al., 2015) was proposed for policy search (Deisenroth et al., 2013), where gradients of the return are typically not available, and, thus, does not exploit gradient information, which would be available in many variational inference settings. MORE fits a quadratic surrogate of the return function, which can be derived from a compatible function approximation perspective and shown to yield *unbiased* natural gradient updates (Pajarinen et al., 2019). Another option is to use the Variational Online Gauss Newton (VOGN) method (Khan & Nielsen, 2018) for optimizing the components. Although VOGN makes use of first-order information, it is mainly popular for mean-field approximations, where it can scale to high-dimensional problems. For lower-dimensional problems (e.g. 100 dimensions), the approximation of the Hessian that VOGN applies can impair the quality of the approximation, in particular, if the target distribution does not decompose into independent likelihoods for different data points.

Instead, we propose a modification of MORE that makes use of the gradient of the log target distribution in order to improve sample efficiency. MORE optimizes a Gaussian distribution by iteratively learning a local quadratic surrogate and updating the Gaussian with respect to this surrogate; our modification of MORE exploits gradient information while fitting the surrogate by penalizing the squared errors between the gradients of the surrogate and the gradients of the log target distribution. Figure 1 illustrates how matching the gradients can lead to better surrogates and, thus, better updates of the variational distribution. In difference to VOGN, the resulting surrogate yields an unbiased approximation of the expected Hessian, resulting in GVAs of higher quality.

Our main contributions are

- presenting a novel method for learning GVAs, gMORE, that incorporates first-order information in MORE,
- reimplementing and adapting VIPS to use different methods for the component optimization, namely VOGN, VON, and GM (Khan & Nielsen, 2018), MORE and gMORE,
- evaluating the different methods for learning GVAs and Gaussian mixture models with respect to sample efficiency and accuracy.

Our modification to MORE is simple, sound and highly effective as we demonstrate in our experiments. Namely, our proposed methods, gMORE and gVIPS, reliably achieve an improved computational efficiency by around one order of magnitude, while retaining the accuracy of their zero-order counterparts, which are state-of-art in the considered problem domain. An open-source implementation for reproducing our experiments is available at *[link will be added here]*.

## 2 PRELIMINARIES

We will now discuss two methods that are essential to this work: MORE (Abdolmaleki et al., 2015) is a stochastic search method that optimizes a Gaussian search distribution, and VIPS (Arenz et al., 2018) is a method for variational inference with GMMs that uses MORE for the component updates.

### 2.1 BLACK BOX VARIATIONAL INFERENCE

Black-box variational inference refers to a specific method (Ranganath et al., 2014) but is here used as an umbrella term for variational inference methods that do not make specific assumptions on the target distribution. These methods aim to minimize the KL $D_{\mathrm{KL}}(q(\boldsymbol{x})||p(\boldsymbol{x}))$ between a model $q(\boldsymbol{x})$ and a target distribution $p(\boldsymbol{x})$ by learning based on samples from the approximation $q(\boldsymbol{x})$ that are evaluated on the target distribution. The optimization problem is typically formulated as follows:

$$\begin{aligned}\arg\min_{q(\boldsymbol{x})} D_{\mathrm{KL}}(q(\boldsymbol{x})||p(\boldsymbol{x})) &= \arg\min_{q(\boldsymbol{x})} \int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})} d\boldsymbol{x} \\ = \arg\min_{q(\boldsymbol{x})} \int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x})} d\boldsymbol{x} + const &= \arg\max_{q(\boldsymbol{x})} \underbrace{\int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \tilde{p}(\boldsymbol{x}) d\boldsymbol{x} + H(q(\boldsymbol{x}))}_{\mathcal{L}(q(\mathbf{x}))}. \end{aligned} \tag{1}$$

The resulting objective $\mathcal{L}(q(\mathbf{x}))$ is known as the evidence lower bound (ELBO) because it bounds the log normalizer of the unnormalized target distribution $\tilde{p}(\boldsymbol{x})$ (which in Bayesian inference corresponds to the log evidence $\log p(\mathcal{D})$). The ELBO can be optimized using gradient descent, where the gradient with respect to the parameters of $q(\boldsymbol{x})$ can be estimated using the log-derivative trick (Ranganath et al., 2014; Williams, 1992) or the reparameterization trick (Rezende et al., 2014; Kingma & Welling, 2014; Titsias & Lázaro-Gredilla, 2014). However, gradient-based optimization can suffer from high variance and poor local optima, in particular for expressive models (Arenz et al., 2020).

#### 2.1.1 MODEL-BASED RELATIVE ENTROPY STOCHASTIC SEARCH

MORE (Abdolmaleki et al., 2015) is a stochastic search method, similar to CEM (Botev et al., 2013), CMA-ES (Shirakawa et al., 2015), or NES (Wierstra et al., 2014), that maximizes a black box function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^N$, by updating a Gaussian search distribution. At every iteration $i$, MORE learns a quadratic surrogate

$$R^{(i)}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{R}^{(i)}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{r}^{(i)} + r^{(i)}, \tag{2}$$

where the symmetric matrix $\boldsymbol{R}^{(i)} \in \mathbb{R}^{N\times N}$, the vector $\boldsymbol{r}^{(i)} \in \mathbb{R}^N$ and the scalar $r^{(i)} \in \mathbb{R}$ are learned using ordinary least squares based on samples $\mathcal{X}^{(i)} \sim q^{(i)}(\boldsymbol{x})$ drawn from the current search distribution $q^{(i)}(\boldsymbol{x})$. Fitting such a surrogate can be derived from a compatible function approximation perspective (Pajarinen et al., 2019; Sutton et al., 2000), showing that MORE yields an unbiased natural gradient update. Intuitively, the surrogate fits the expected Hessian and gradient of the optimization objective. MORE uses this surrogate to update the search distribution subject to constraints on its entropy $H(q(\boldsymbol{x}))$ and its KL divergence to the last search distribution, $D_{\mathrm{KL}}(q(\boldsymbol{x})||q^{(i)}(\boldsymbol{x}))$,

$$q^{(i+1)}(\boldsymbol{x}) = \arg\max_{q(\boldsymbol{x})} \int_{\boldsymbol{x}} q(\boldsymbol{x}) R^{(i)}(\boldsymbol{x}) d\boldsymbol{x}, \quad \text{st. } H(q(\boldsymbol{x})) \geq \beta, \quad D_{\mathrm{KL}}(q(\boldsymbol{x})||q^{(i)}(\boldsymbol{x})) \leq \epsilon, \tag{3}$$

where $\epsilon$ is a hyperparameter relating to a step size and $\beta$ is typically decreased at every iteration for annealing. The purpose of the entropy constraint is to maintain exploration during optimization, and the KL constraint should force the updated search distribution to stay in the validity of the quadratic surrogate $R^{(i)}$. Thanks to the quadratic structure of the surrogate $R^{(i)}$ the solution of optimization problem 3 is Gaussian, with mean $\boldsymbol{\mu}^{(i+1)}$ and covariance $\boldsymbol{\Sigma}^{(i+1)}$ given by

$$\boldsymbol{\Sigma}^{(i+1)} = \left(\frac{\eta}{\eta+\omega}{\boldsymbol{\Sigma}^{(i)}}^{-1} - \frac{1}{\eta+\omega}\boldsymbol{R}^{(i)}\right)^{-1}, \boldsymbol{\mu}^{(i+1)} = \boldsymbol{\Sigma}^{(i+1)}\left(\frac{\eta}{\eta+\omega}{\boldsymbol{\Sigma}^{(i)}}^{-1}\boldsymbol{\mu}^{(i)} + \frac{1}{\eta+\omega}\boldsymbol{r}^{(i)}\right), \tag{4}$$

where $\eta \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$ are the Lagrangian multipliers corresponding to the entropy and KL constraint, that can be efficiently found by minimizing the convex Lagrangian dual function.

Although MORE treats $q(\boldsymbol{x})$ merely as a search distribution that converges to a singular distribution during optimization in order to find optimal parameters $\boldsymbol{x}$, Arenz et al. (2018) showed that MORE can also be used for learning Gaussian approximations for variational inference (where we have direct interest in $q(\boldsymbol{x})$), by dropping the entropy constraint in favor of an entropy bonus in the objective function with fixed weight of 1. This minor modification of MORE substitutes $\omega = 1$ when updating mean $\boldsymbol{\mu}^{(i+1)}$ and covariance $\boldsymbol{\Sigma}^{(i+1)}$ and only optimizes the stepsize parameter $\eta$.

### 2.2 Variational Inference by Policy Search

VIPS (Arenz et al., 2018; 2020) optimizes a Gaussian mixture model

$$q(\boldsymbol{x}) = \sum_o q(o) q(\boldsymbol{x}|o)$$

for variational inference, where $q(o)$ is a categorical distribution that assigns weight to each Gaussian component $q(\boldsymbol{x}|o)$. Arenz et al. (2018) introduced an auxiliary distribution $\tilde{q}(o|\boldsymbol{x})$ to derive a lower bound $L(q(\boldsymbol{x}), \tilde{q}(o|\boldsymbol{x}))$ on the ELBO objective,

$$\begin{aligned} L(q(\boldsymbol{x}), \tilde{q}(o|\boldsymbol{x})) &= \sum_o q(o) \Big[ \overbrace{\int_{\boldsymbol{x}} q(\boldsymbol{x}|o) \left(\log \tilde{p}(\boldsymbol{x}) + \log \tilde{q}(o|\boldsymbol{x})\right) d\boldsymbol{x} + H(q(\boldsymbol{x}|o))}^{J_o(q(\boldsymbol{x}|o))} \Big] + H(q(o)) \quad (5) \\ &= \int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \tilde{p}(\boldsymbol{x}) d\boldsymbol{x} + H(q(\boldsymbol{x})) - \sum_{\boldsymbol{x}} q(\boldsymbol{x}) D_{\mathrm{KL}}(q(o|\boldsymbol{x}) || \tilde{q}(o|\boldsymbol{x})) d\boldsymbol{x} \\ &\leq \int_{\boldsymbol{x}} q(\boldsymbol{x}) \log \tilde{p}(\boldsymbol{x}) d\boldsymbol{x} + H(q(\boldsymbol{x})). \end{aligned}$$

For a given auxiliary distribution $\tilde{q}(\boldsymbol{x})$ maximizing the lower bound $L(q(\boldsymbol{x}), \tilde{q}(o|\boldsymbol{x}))$ is significantly easier than maximizing the ELBO, because every component can be optimized independently by maximizing $J_o(q(x|o))$. To ensure improvement on the ELBO, VIPS employs a similar procedure as expectation maximization (Bishop, 2006)—which minimizes the forward KL $D_{\mathrm{KL}}(p(\boldsymbol{x})||q(\boldsymbol{x}))$—, by exploiting that the bound is tight when the auxiliary distribution $\tilde{q}(o|\boldsymbol{x})$ is set to the true responsibilities $q(o|\boldsymbol{x})$. Namely, at every iteration $i$, the responsibilities are computed based on the current approximation $q^{(i)}(\boldsymbol{x})$ and then held constant for optimizing the lower bound for each component and for the distribution of the weights independently. Hence, after every iteration an improvement on the lower bound $L(q(\boldsymbol{x}), \tilde{q}(o|\boldsymbol{x}))$ also ensures an improvement on the original objective.

For updating the individual Gaussian component by increasing $J_o(q(\boldsymbol{x}|o))$ (see Eq. 5), VIPS uses a slight variation of MORE with fixed entropy bonus, as discussed in Section 2.1.1, where the quadratic surrogate approximates the component-specific objective function $f_o(\boldsymbol{x}) = \log \tilde{p}(\boldsymbol{x}) + \log \tilde{q}(o|\boldsymbol{x})$. In this work, we will also investigate other methods for optimizing the component, namely VON (Khan & Nielsen, 2018) or VOGN (Khan & Nielsen, 2018), GM (Khan & Nielsen, 2018) and our new method gMORE. We will evaluate the different options in Section 5.

For updating the weights $q(o)$, VIPS use Monte-Carlo estimates $\tilde{J}_o(q(x|o))$ of $J_o(q(x|o))$ to compute the optimal update in closed form, namely $q^{(i+1)}(o) \propto \exp(\tilde{J}_o(q(x|o)))$.

Arenz et al. (2020) also discuss several details such as an adaptation scheme that dynamically adds and deletes components, and importance weighting to reuse previous function evaluations. While we also apply these techniques, they are orthogonal to our contribution, which modifies MORE to use gradients, and, hence, we kindly refer to the original article (Arenz et al., 2020).

## 3 Unbiased First-Order Natural Gradient Updates

We will now discuss the main technical contribution of our work, *gMORE*, a modification of MORE that can make use of gradient information to increase sample efficiency. *gMORE* can be used as drop-in replacement in VIPS, to obtain *gVIPS*, a state of the art method for learning GMMs for VI.

### 3.1 INTERLUDE: FITTING THE SURROGATE WITHOUT GRADIENTS

At every iteration, MORE (Abdolmaleki et al., 2015) learns a compatible quadratic surrogate $R^{(i)}(\boldsymbol{x})$ (Eq. 2), to fit the objective function $f(\boldsymbol{x})$ at locations $\boldsymbol{x}_s \sim q(\boldsymbol{x})$ sampled from the current search distribution. The surrogate is linear in its parameters, that is, it can be written as

$$R_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\theta},$$

with parameters $\boldsymbol{\theta}$ and corresponding features $\boldsymbol{\phi}(\boldsymbol{x})$

$$\begin{aligned} \boldsymbol{\theta} &= \left[R_{1,1}, R_{1,2}, \ldots, R_{1,N}, R_{2,2}, R_{2,3}, \ldots R_{2,N}, \ldots, R_{N,N}, r_1, \ldots, r_N, r\right]^\top \\ \boldsymbol{\phi}(\boldsymbol{x}) &= \left[0.5x_1^2, x_1x_2, \ldots, x_1x_N, 0.5x_2^2, x_2x_3, \ldots x_2x_N, \ldots, 0.5x_N^2, x_1, \ldots, x_N, 1\right]^\top. \end{aligned} \tag{6}$$

Hence, the parameters $\boldsymbol{\theta}^\star$ of the surrogate can be learned in closed form using ordinary least-squares,

$$\boldsymbol{\theta}^\star = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y}, \tag{7}$$

where each row of $\boldsymbol{\Phi}$ contains the features of the respective sample and each row of $\boldsymbol{y}$ contains the corresponding function evaluation. This procedure, which is used by MORE (Abdolmaleki et al., 2015; Arenz et al., 2018), minimizes the loss function

$$L_{\text{OLS}}(\boldsymbol{\theta}) = \sum_{s=1}^{N_s} \left(R_{\boldsymbol{\theta}}(\boldsymbol{x}_s) - f(\boldsymbol{x}_s)\right)^2,$$

where $N_s$ is the number of samples. In practice, we use ridge regression for regularization.

### 3.2 FITTING THE SURROGATE WITH GRADIENTS

In order to make use of gradient information for fitting the surrogate $R_{\boldsymbol{\theta}}(\boldsymbol{x})$, we minimize the loss

$$L_{\text{gOLS}}(\boldsymbol{\theta}) = \sum_{s=1}^{N_s} \left[ \left(R_{\boldsymbol{\theta}}(\boldsymbol{x}_s) - f(\boldsymbol{x}_s)\right)^2 + \sum_{i=1}^{N} \left( \frac{\partial R_{\boldsymbol{\theta}}}{\partial x_i}(\boldsymbol{x}_s) - \frac{\partial f}{\partial x_i}(\boldsymbol{x}_s) \right)^2 \right],$$

that is, we also want to match the partial derivatives $\frac{\partial f}{\partial x_i}$ of the objective function $f(\boldsymbol{x})$.

The gradient $\nabla_{\boldsymbol{x}} R(\boldsymbol{x}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{r}$ of the quadratic surrogate is also linear in the parameters $\boldsymbol{\theta}$, and hence, we can write the loss as

$$L_{\text{gOLS}}(\boldsymbol{\theta}) = \sum_{s=1}^{N_s} \left[ \left(\boldsymbol{\phi}(\boldsymbol{x}_s)^\top \boldsymbol{\theta} - f(\boldsymbol{x}_s)\right)^2 + \left(\boldsymbol{\Phi}^g(\boldsymbol{x}_s)\boldsymbol{\theta} - \nabla_x f(\boldsymbol{x}_s)\right)^\top \left(\boldsymbol{\Phi}^g(\boldsymbol{x}_s)\boldsymbol{\theta} - \nabla_x f(\boldsymbol{x}_s)\right) \right],$$

where the gradient-feature matrix $\boldsymbol{\Phi}^g(\boldsymbol{x})$ is a sparse matrix that is constructed such that its matrix product with the parameter vector corresponds to the gradient of $R(\boldsymbol{x})$, that is, $\boldsymbol{\Phi}^g(\boldsymbol{x}_s)\boldsymbol{\theta} = \nabla_x R(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_s}$. The gradient-feature matrix $\boldsymbol{\Phi}^g$ has a recursive structure, which is illustrated for four dimensions in Figure 2. As the only non-zero elements are either an element of $\boldsymbol{x}$ or 1, we can compute the matrix efficiently by precomputing the indices for each of these elements.

Constructing the design matrix $\boldsymbol{\Phi}$ by stacking zero-order features $\boldsymbol{\phi}(\boldsymbol{x}_s)^\top$ and first-order features $\boldsymbol{\Phi}^g(\boldsymbol{x}_s)$ for each sample $\boldsymbol{x}_s$, and constructing the targets $\boldsymbol{y}$ by stacking the respective function evaluations $f(\boldsymbol{x}_s)$ and gradients $\nabla f(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_s}$, that is,

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^\top \\ \boldsymbol{\Phi}^g(\boldsymbol{x}_1) \\ \boldsymbol{\phi}(\boldsymbol{x}_2)^\top \\ \boldsymbol{\Phi}^g(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_{N_s})^\top \\ \boldsymbol{\Phi}^g(\boldsymbol{x}_{N_s}) \end{pmatrix} \qquad \boldsymbol{y} = \begin{pmatrix} f(\boldsymbol{x}_1) \\ \nabla f(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_1} \\ f(\boldsymbol{x}_2) \\ \nabla f(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_2} \\ \vdots \\ f(\boldsymbol{x}_{N_s}) \\ \nabla f(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_{N_s}} \end{pmatrix},$$

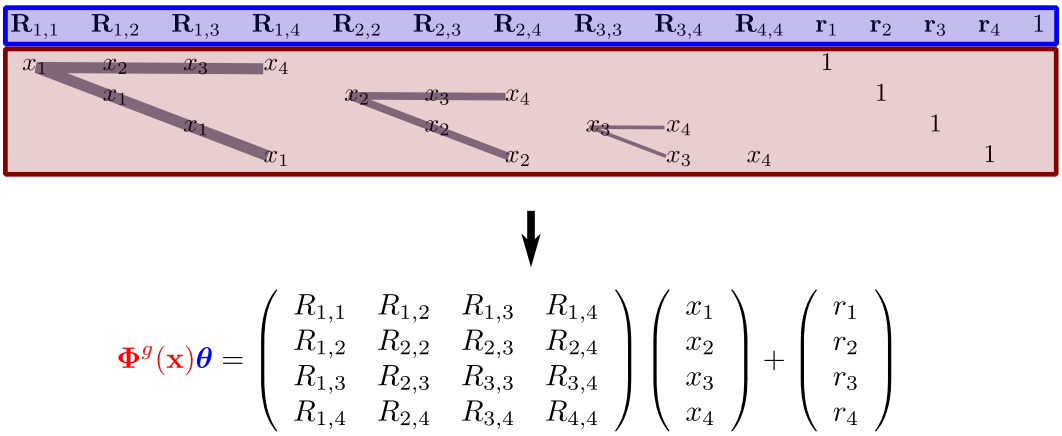



Figure 2: The top row (shaded in blue) shows the parameter vector $\boldsymbol{\theta}^\top$ of the quadratic surrogate for $N = 4$. The lower matrix (shaded in red) shows the non-zero elements of the gradient-feature matrix $\mathbf{\Phi}^g(\boldsymbol{x})$, such that the matrix product $\mathbf{\Phi}^g(\boldsymbol{x})\boldsymbol{\theta}$ corresponds to the gradient of $R(\boldsymbol{x})$ (Eq. 2). The V-shapes highlight the recursive structure of $\mathbf{\Phi}^g(\boldsymbol{x})$.

we can express our loss in matrix notation as a standard least squares problem,

$$L_{\text{gOLS}}(\boldsymbol{\theta}) = \Big(\mathbf{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\Big)^\top \Big(\mathbf{\Phi}\boldsymbol{\theta} - \boldsymbol{y}\Big),$$

with a closed-form solution given by Eq. 7. However, in our experiments we use weighted ridge regression, in order to reuse samples from previous iterations as discussed by Arenz et al. (2020), and hence the optimal parameters for the surrogate are

$$\boldsymbol{\theta}^\star = (\mathbf{\Phi}^\top \boldsymbol{W} \mathbf{\Phi} + \delta \boldsymbol{I}_n)^{-1} \mathbf{\Phi}^\top \boldsymbol{W} \boldsymbol{y},$$

with a diagonal weighting matrix $\boldsymbol{W}$ assigning different weights to different samples, and ridge coefficient $\delta$ penalizing the $\ell_2$-norm of the parameters.

## 4 Related Work

Early uses of the Gaussian variational approximation (GVA) (Opper & Archambeau, 2009) date back to the last century. For example, Barber & Bishop (1998) learned a GVA for small Bayesian neural networks and Seeger (2000) learned a GVA for approximating the posterior of the hyperparameters of a support vector machine. Slightly more recently, Graves (2011) related the ELBO to the minimum description length and optimized a mean-field GVA. Challis & Barber (2013) derived conditions on the likelihood terms for which the ELBO objective for GVAs is smooth and concave.

### 4.1 Optimizing GVAs

A rather simple way to optimize GVAs is to use gradient descent, however, estimating the gradient of the ELBO is not straightforward. Ranganath et al. (2014) proposed a black-box optimizer that only needs access to the target function. They apply the log-derivative (or REINFORCE-) trick (Williams, 1992) to obtain unbiased estimates which, however, suffer from high variance. Several researchers (Rezende et al., 2014; Kingma & Welling, 2014; Titsias & Lázaro-Gredilla, 2014) proposed a reparameterization for obtaining lower-variance estimates of the gradient, which requires the gradients of the target function. Sakaya et al. (2017) show how importance sampling can be used to estimate the reparameterization gradients for GVAs from previous samples.

Second-order methods have been investigated by Fan et al. (2015) and Regier et al. (2017). Fan et al. (2015) propose a second-order method that uses reparameterization of the Gaussian to obtain unbiased estimates of the second-order derivative of the ELBO based on second-order derivatives of the log target distribution. The Hessian can then be used to compute an update using Newton's method. Regier et al. (2017) extend their idea by introducing a Euclidean trust region.

Currently, the probably most efficient approaches are based on natural gradient descent (Amari, 1998). Khan et al. (2015) proposed a KL-proximal method for VI and showed that their update is equivalent to natural gradients. Using a linearization of the ELBO their method was applied for learning GVAs for non-conjugate models. Khan et al. (2016) extended this idea to other divergences and stochastic gradients. Khan & Nielsen (2018) proposed a more direct estimation of the natural gradient, VON, that uses samples of the Hessian of the target distribution to obtain unbiased estimates of the natural gradient. They also proposed a first-order method, VOGN, that estimates the Hessian using the generalized Gauß-Newton approximation, which, however, results in biased estimates. VOGN is highly related to the scope of this article, as it mainly differs from gMORE in the way the expected Hessian and the expected gradient are estimated. We provide a more detailed description of VON and VOGN in Appendix A.5, where we also show the close relation between VON, VOGN and (g)MORE. Salimbeni et al. (2018) compute the natural gradient based on the Jacobian of the parameters of the Gaussian and its expectation parameters. The Jacobian can be computed using forward-mode differentiation, or by using reverse-mode differentiation twice.

## 5 Experiments

We will now evaluate how the incorporation of gradient information affects the quality and sample efficiency of MORE for learning GVAs. We will also compare our method with VOGN, and VON, which assumes access to the Hessian of the target distribution and acts as our baseline. However, VOGN is not always applicable as it assumes a typical posterior structure (see Equation 11 in Appendix A.5). One may argue that any target distribution $\tilde{p}(\boldsymbol{x})$ is a special case of that structure, using a single (virtual) data point for the target likelihood and a uniform prior. We will also evaluate this variant, which is related to the gradient magnitude (GM) method (Khan & Nielsen, 2018).

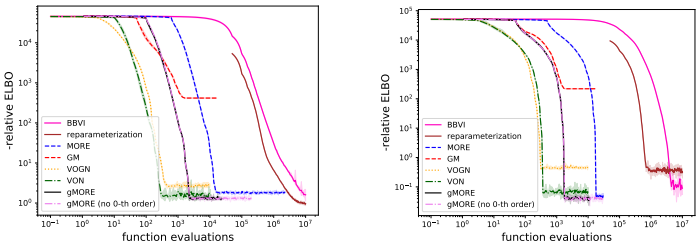
As our primary focus is on learning highly accurate multi-modal distribution, we perform most experiments in the GMM setting, where we evaluate how the different options for optimizing GVAs affect the performance of VIPS. We reimplemented VIPS, which was only available in C++, in Tensorflow (Abadi et al., 2015) and investigated the following variants: VIPS (which uses MORE), gVIPS, vonVIPS, vognVIPS and gmVIPS. To allow for a fair comparison with the original VIPS, we acquired the data from Arenz et al. (2020) and evaluate our methods on the same experiments. Hence, we can also relate our results to the original implementation of VIPS (which we call VIPS++) as well as many MCMC and VI methods that have been tested by Arenz et al. (2020).

- In *German credit* and *breast cancer* we want to approximate the posterior for Bayesian logistic regression based on the respective data sets (Lichman, 2013). *German credit* has 25 parameters and 1000 data points; *breast cancer* uses 31 parameters and 561 data points.
- *Planar robot* considers the problem of sampling joint configurations of planar robot with 10 links, that aims to reach a given goal position with a smooth configuration. We test both variants that were used by Arenz et al. (2020), one with a single goal position and one, where the robot can choose among four different goal positions. Visualizations of the robot and the learned solution can be found in Appendix A.7.
- In the *Target GMM* experiment, we use the log-density function of a GMM with ten components as target distribution, but do not provide the parameters to the algorithm. Approximating the GMM is a hard exploration problem, as the different components have little overlap with each other. In contrast to Arenz et al. (2020) we do not only investigate 20-, 40- and 60-dimensional GMMs, but also 80 and 100 dimensions as we noticed that our methods can also tackle these higher-dimensional problems.

For the exact specification of the target distributions, we kindly refer to the original article (Arenz et al., 2020). Implementation details, as well as hyperparameters, can be found in Appendix A.6. In the following plots, we show the negated ELBO on log-log plots over function evaluations. We subtract a constant offset to ensure positive values. The offsets can be found in Appendix A.6.4.

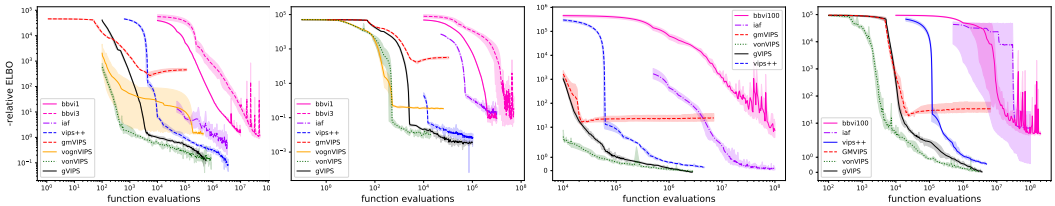
### 5.1 Learning a Gaussian Variational Approximation

In the first experiment, we optimize a single Gaussian approximation on the *breast cancer* and *German credit* test problem. We evaluate MORE, gMORE, VOGN, VON, GM, and—as an ablation—a




(a) Breast Cancer (b) German Credit

Figure 3: In our GVA experiments, using the vanilla gradient, computed with the log-derivative or reparameterization trick, can learn good approximations but is very sample inefficient. MORE, gMORE and VON achieve similar approximation quality, but high-order methods are more sample efficient. VOGN is fast, but its approximation quality suffers from the bias of the Gauß-Newton approximation, which is amplified when applying it to the target log density directy (GM).




(a) Breast Cancer (b) German Credit (c) Planar Robot (1 goal) (d) Planar Robot (4 goals)

Figure 4: The result when learning GMMs are similar to the GVA experiment. However, the bias of VOGN has a stronger effect, and gMORE is able to catch up with VON during the optimization.

variant of gMORE that does not make use of zero order information. As shown in Figure 3, MORE, gMORE and VON learn similarly well approximations, which was expected, as all these methods use unbiased estimates of the natural gradient. However, as the methods exploit different amounts of information, they differ in terms of sample efficiency: The second-order method VON is around one order of magnitude more efficient than the first-order method gMORE, which in term is one order of magnitude more efficient than the zero-order method MORE. VOGN also achieved high sample efficiency, similar to VON, however at the cost of converging to worse approximations, which we explain by the bias introduced by the generalized Gauß-Newton approximation. The limitations of this approximation are particularly apparent when considering the performance of GM, which applies it in black-box fashion to the likelihood of the complete training data. Our ablation without zero order information performs indistinguishable from gMORE, highlighting the value of the additional first-order information that is exploited by gMORE. We also evaluated the different methods with respect to wallclock time and found that MORE, gMORE and VON perform very similar. We show the learning curves in Appendix A.4, were we also justify our focus on sample efficiency, and the first-order setting in particular.

### 5.2 Optimizing a GMM

In the second experiment we evaluate the performance of the different methods, when they are used within the inner optimization of VIPS. We perform five experiments namely breast cancer, German credit, GMM, planar robot (with one goal and with four goals). Here, we only show the evaluations with respect to the ELBO. Evaluations with respect to the maximum mean discrepancy (Gretton et al., 2012) can be found in Appendix A.3, where we also show the performance of many MCMC methods that were tested by Arenz et al. (2020). The results are shown in Figure 4. On the *Target GMM* experiment, we only tested VIPS, gVIPS and vonVIPS, as gmVIPS was not able to solve the task. All of the tested methods learned indisguishable good approximations with estimated KLs below $1e-7$. The required samples for reaching this threshold are shown in Table 1. Overall, the VIPS results are in line with the results for single GVAs. Namely, MORE, gMORE and VON achieve similarly good approximations, where higher-order methods achieve better sample efficiency. However,

Table 1: Average samples required to solve the GMM experiments

| Method | 20 D | 40 D | 60 D | 80 D | 100 D |
|---|---|---|---|---|---|
| VIPS | $1.19e5 \pm 6e4$ | $2.29e5 \pm 1e4$ | $6.48e5 \pm 9e4$ | N/A | N/A |
| gVIPS | $15541 \pm 2098$ | $28796 \pm 4610$ | $48072 \pm 5670$ | $87037 \pm 8383$ | $310923 \pm 90784$ |
| vonVIPS | $19243 \pm 4129$ | $18403 \pm 1452$ | $23996 \pm 1743$ | $38068 \pm 7786$ | $39894 \pm 13800$ |

VOGN is not able to exploit the increased expressiveness of the variational approximation, likely due to the bias resulting from the generalized Gauß-Newton method. It is also interesting to note, that while VON still enjoys one order of magnitude better sample efficiency in the beginning, gMORE is able to catch up during optimization. We believe that for the highly multi-modal *planar robot* experiments, exploration is a bottleneck, where the improved sample efficiency from second-order information is not helpful. For the approximately unimodal *breast cancer* and *German credit* experiments, all components of the GMM are highly overlapping, and hence, can make use of each others samples; the better sample efficiency for optimizing a single component, might therefore diminish.

## 6 Conclusion

Although MORE was developed as a stochastic search method, it was recently shown to be effective also for learning GVAs for variational inference—in particular when it is used as subroutine for optimizing GMMs. However, as a black-box optimizer, MORE does not make use of gradient information which is wasteful in variational inference. We presented a simple technique to address this limitation using a modification to least-squares that also matches the target gradient when fitting the quadratic surrogate. The resulting method, gMORE, is highly effective. Compared to MORE it is about one order magnitude more efficient. Unlike VOGN, it does not suffer from biases caused by the generalized Gauß-Newton approximation, and unlike VON it does not assume access to the Hessian. We also proposed to use different component optimizers in the inner optimization of VIPS. We tested VON, VOGN, GM and gMORE against the previously proposed MORE. We found that VON is slightly preferable compared to gMORE when the Hessian matrix is available, although its increased sample efficiency typically diminishes during optimization. If the Hessian is not available, gMORE is preferable to VOGN as it does not suffer from biases that impair the quality of the learned approximation. gMORE is also around one order of magnitude more efficient than MORE, and hence MORE should only be used when the target distribution is not differentiable.

### 6.1 Limitations and Future Work

While GVAs with full covariance can yield significantly better approximations than mean-field approximations they do not scale to very high dimensions, as the matrix inversion becomes too costly and the memory footprint too large. In this work, we do not consider very high-dimensional problems, but instead focus on learning highly accurate approximations for medium-scaled problems (below 100 dimensions). Possible applications are for example Bayesian inference with medium-sized models (Thijssen & Wessels, 2020), and robotics applications, such as inverse kinematics (Pignat et al., 2020) or path planning (Ewerton et al., 2020) problems. However, we acknowledge that there is a high demand for variational inference for high-dimensional problems, e.g. Bayesian inference for deep networks. Several methods (Barber & Bishop, 1998; Seeger, 2000; Maddox et al., 2019; Tomczak et al., 2020; Mishkin et al., 2018) reduced the trainable parameters of GVAs by using factor analyzed covariance, that is, covariance of the form $\boldsymbol{\Sigma} = \boldsymbol{A}^\top \boldsymbol{A} + \boldsymbol{I}\boldsymbol{d}$, where $\boldsymbol{A} \in \mathbb{R}^{L \times N}$ contains $L < N$ factors and $\boldsymbol{d}$ are diagonal offsets to ensure that $\boldsymbol{\Sigma}$ has full rank. We believe that such structure could also be used by gMORE. However, as a low rank surrogate would not be linear in the parameters, we will need to use gradient descent instead of ordinary least squares for optimizing our surrogate objective. Furthermore, the natural gradient update can double the amount of factors, which we could counter using a projection based on the largest eigenvalues, as proposed by Mishkin et al. (2018) for SLANG, which is based on VOGN.

## REFERENCES

M. Abadi, M. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

A. Abdolmaleki, R. Lioutikov, N. Lua, L. Paulo Reis, J. Peters, and G. Neumann. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 153–154, 2015.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

O. Arenz, M. Zhong, and G. Neumann. Efficient gradient-free variational inference using policy search. In *International Conference on Machine Learning (ICML)*, 2018.

Oleg Arenz, Mingjun Zhong, and Gerhard Neumann. Trust-region variational inference with gaussian mixture models. *Journal of Machine Learning Research*, 21(163):1–60, 2020. URL `http://jmlr.org/papers/v21/19-524.html`.

Simon R Arridge, Kazufumi Ito, Bangti Jin, and Chen Zhang. Variational gaussian approximation for poisson data. *Inverse Problems*, 34(2):025005, jan 2018. doi: 10.1088/1361-6420/aaa0ab. URL `https://doi.org/10.1088/1361-6420/aaa0ab`.

David Barber and Christopher M Bishop. Ensemble learning for multi-layer networks. *Advances in neural information processing systems*, pp. 395–401, 1998.

Philipp Becker, Oleg Arenz, and Gerhard Neumann. Expected information maximization: Using the i-projection for mixture density estimation. In *International Conference on Learning Representations*, 2019.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, 2006.

Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In *Handbook of statistics*, volume 31, pp. 35–59. 2013.

Edward Challis and David Barber. Gaussian kullback-leibler approximate inference. *Journal of Machine Learning Research*, 14(32):2239–2286, 2013. URL `http://jmlr.org/papers/v14/challis13a.html`.

M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, pp. 388–403, 2013.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Marco Ewerton, Oleg Arenz, and Jan Peters. Assisted teleoperation in changing environments with a mixture of virtual guides. *Advanced Robotics*, 34(18):1157–1170, 2020. doi: 10.1080/01691864.2020.1785326.

Kai Fan, Ziteng Wang, Jeff Beck, James Kwok, and Katherine Heller. Fast second-order stochastic backpropagation for variational inference. *arXiv preprint arXiv:1509.02866*, 2015.

S. J. Gershman, M. D. Hoffman, and D. M. Blei. Nonparametric variational inference. In *International Conference on Machine Learning (ICML)*, 235–242, 2012.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13:723–773, March 2012. ISSN 1532-4435.

Mohammad Emtiyaz Khan and Didrik Nielsen. Fast yet simple natural-gradient descent for variational inference in complex models. In *2018 International Symposium on Information Theory and Its Applications (ISITA)*, pp. 31–35. IEEE, 2018.

Mohammad Emtiyaz Khan, Pierre Baqué, François Fleuret, and Pascal Fua. Kullback-leibler proximal variational inference. In *Proceedings of the international conference on Neural Information Processing Systems*, number CONF, 2015.

Mohammad Emtiyaz Khan, Reza Babanezhad, Wu Lin, Mark Schmidt, and Masashi Sugiyama. Faster stochastic variational inference using proximal-gradient methods with general divergence functions. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 319–328, 2016.

D. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4743–4751, 2016.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2378–2386. Curran Associates, Inc., 2016.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32:13153–13164, 2019.

Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark W Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *NeurIPS*, 2018.

I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 541–548, 2010.

R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 06 2003. doi: 10.1214/aos/1056562461.

Manfred Opper and Cédric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

J. Pajarinen, H.L. Thai, R. Akrour, J. Peters, and G. Neumann. Compatible natural gradient policy search. (8):1443–1466, 2019.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

Emmanuel Pignat, Teguh Lembono, and Sylvain Calinon. Variational inference with mixture model approximation for applications in robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3395–3401. IEEE, 2020.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Jeffrey Regier, Michael I Jordan, and Jon McAuliffe. Fast black-box variational inference through stochastic trust-region optimization. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2399–2408, 2017.

D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, pp. 1278–1286, 2014.

Joseph Sakaya, Arto Klami, et al. Importance sampled stochastic optimization for variational inference. In *33rd Conference on Uncertainty in Artificial Intelligence 2017 Sydney, Australia, 11-15 August 2017*. AUAI Press, 2017.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *International Conference on Artificial Intelligence and Statistics*, pp. 689–697. PMLR, 2018.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

Matthias Seeger. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *Proceedings of the 13th Annual Conference on Neural Information Processing Systems*, pp. 603–609, 2000.

S. Shirakawa, Y. Akimoto, K. Ouchi, and K. Ohara. Sample reuse in the covariance matrix adaptation evolution strategy based on importance sampling. In *Annual Conference on Genetic and Evolutionary Computation*, pp. 305–312, 2015.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Bram Thijssen and Lodewyk FA Wessels. Approximating multivariate posterior distribution functions from monte carlo samples for sequential bayesian inference. *PloS one*, 15(3):e0230101, 2020.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pp. 1971–1979. PMLR, 2014.

Marcin Tomczak, Siddharth Swaroop, and Richard Turner. Efficient low rank gaussian variational inference for neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4610–4622. Curran Associates, Inc., 2020.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

# A Appendix

## A.1 Ethics Statement

We believe wholeheartedly that researchers must take responsibility for their research and that we cannot rely on others to use our results only for the benefit of society or on governments to force them to do so. We must constantly question how our results are actually being used. We also cannot hide behind the fact that the actual impact of our research is likely to be small, because progress is usually made in many small steps, and if we did not believe in our own footprints, it would be hard to justify our research at all. Machine learning methods are extremely disruptive and have strong implications for our daily lives. They have the potential to relieve our burden and improve our quality of life, help us address challenges such as the climate crisis, or lead to important advances in various fields, such as medicine. However, machine learning methods also carry many risks that are already emerging: they can reinforce prejudices, discriminate against people, invade our privacy,

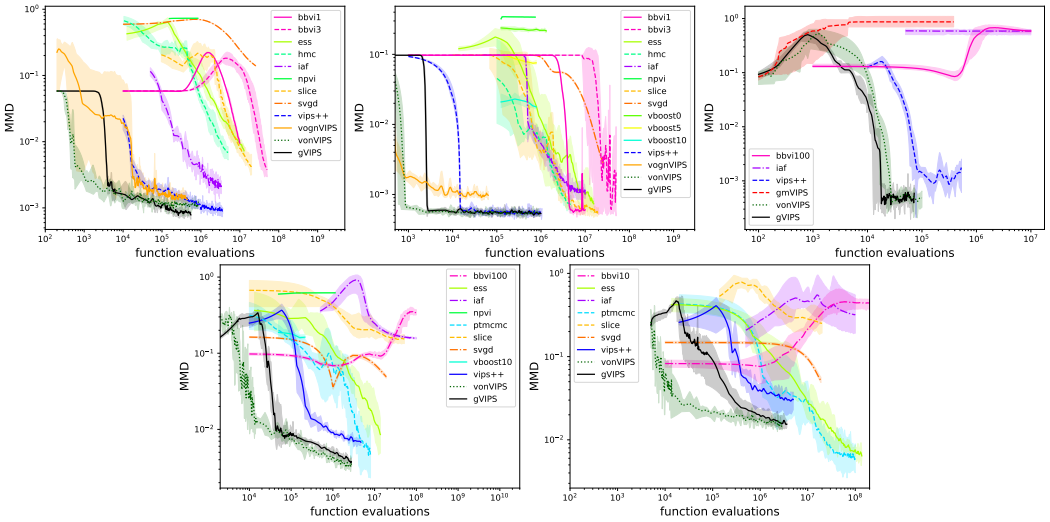



Figure 5: From left to right: BreastCancer, GermanCredit, GMM(20D), Planar Robot (1 goal), Planar Robot (4 goals)

waste huge amounts of energy, or increase inbalances in power and wealth. They can cause serious harm—even fatal accidents—if we overestimate their capabilities, and they can be used maliciously, for example, to forge data or carry out cyberattacks.

How should this work be framed in terms of all these positive and negative impacts? We believe that wasting energy is a relatively small risk, because in contrast to most deep learning method, we focus on structured representations with few parameters that can be efficiently learned. Similarly, we are less vulnerable to the risk of privacy invasion or discrimination compared to many computer vision and NLP methods. Regarding the other opportunities and risks, we can not assess in which directions our small steps leads most, but we are positive that we can use our insights to assist humans in their daily life.

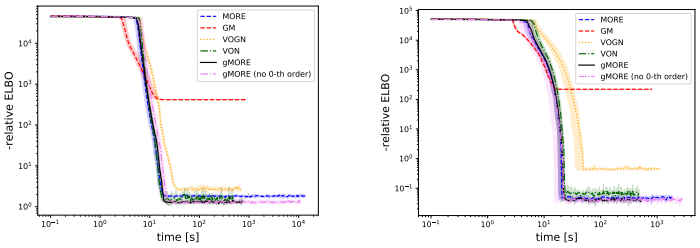
### A.2 Reproducibility Statement

We also believe wholeheartedly that reproducibility, transparency, and openness are essential in research and that we could make much faster progress if data and code were shared more consequently and limitations and negative results were better communicated. For this reason, we provide a detailed description of our implementation in Appendix A.6, where we also provide the hyperparameters for all of our experiments and the procedure we used to select them. Of course, we will open source the implementation we used for our experiments, and of course we will make the anonymized code available to reviewers. Our code is prepared in such a way that any combination of test problem and method we have implemented can be easily executed from the command line, presetting the hyperparameters we used for our evaluations.

### A.3 MMD Evaluations

We also compared our VIPS variants to the MCMC and VI methods that were tested by Arenz et al. (2020). This evaluation is based on the maximum mean discrepancy (MMD) (Gretton et al., 2012), approximate samples and groundtruth samples, since evaluating the ELBO is not possible for MCMC methods. We computed the MMD in the same way as described by Arenz et al. (2020) and also use the same groundtruth samples, which were created by very long MCMC runs. Please refer to the original work for details (Arenz et al., 2020). Figure 5 shows evaluations for the following additional methods:

- BBVI (log-derivative trick) with 1 component and 3 components (bbvi1 and bbvi3, Ranganath et al. 2014)
- Elliptical Slice Sampling (ess, Murray et al. 2010),




(a) Breast Cancer (b) German Credit

Figure 6: The different approaches to estimate the natural gradient perform remarkably similar when evaluated with respect to wallclock time. However, we argue that this does not justify the conclusion that it does not matter whether zero-order, first-order or second-order methods are applied, since we only investigated problems where the target distribution (and its gradient and Hessian) can be evaluated efficiently, which is in general not the case.

- Hybrid (or Hamiltonian) Monte Carlo (hmc, Duane et al. 1987),
- Inverse Autoregressive Flows (iaf, Kingma et al. 2016),
- Non-Parametric Variational Inference (npvi, Gershman et al. 2012),
- Slice Sampling (slice, Neal 2003),
- Stein-Variational Gradient Descent (svgd, Liu & Wang 2016).

For the VIPS variants tested by us, the approximation quality with respect to the MMD is very similar to the evaluation of the ELBO. The comparison to MCMC and other VI methods, highlights the usefulness of GMM variational approximations.

### A.4 Evaluation with Respect to Wallclock Time

We also evaluated our GVA experiments with respect to wallclock time and present the results in Figure 6. Here, VON, VOGN, MORE and gMORE[1] perform remarkably similar. However, our experiments did not consider target distributions that are costly to evaluate, for example when the random variable $\boldsymbol{x}$ correspond to a hyperparameter of a neural network that needs to be trained in order to evaluate its quality, or if it relates to a trajectory-parameterization of a robot, which needs to be simulated. In such settings, the improved sample efficiency of higher-order methods will certainly also be reflected in the wallclock time. Indeed, we argue that the first-order setting, which we focused on in this work, is the most important setting for variational inference, as gradients of the target distributions are typically available and are usually much more efficient than zero-order methods. Furthermore, second-order methods scale poorly to high dimensions, as the evaluation of the Hessian becomes too costly. While we do not consider very high dimensional problems in this work due to the intractability of full covariance Gaussians, we plan to use factory analyzers to tackle this problem setting, as discussed in Section 6.1. We believe that our first-order method for obtaining unbiased estimates of the natural gradient, is well-suited for this problem class of problems.

### A.5 Relation between VOGN, VON and (g-)MORE

VON (Khan & Nielsen, 2018) optimizes the Gaussian variational approximation using natural gradient descent. Intuitively, the natural gradient $\tilde{\nabla}$ points in the direction of steepest ascend when making small changes to the distribution (rather than to its parameters which leads to the vanilla gradient $\nabla$) and hence typically leads to much more efficient updates (Amari, 1998). The natural gradient can be computed by prescaling the vanilla gradient with the inverse of the Fisher information matrix. However, Khan & Nielsen (2018) presented a simpler method for estimating the natural gradient for learning GVAs, which exploits that for exponential-family distributions, the natural gradient of natural parameters coincides with the vanilla gradient of the expectation parameters. By

[1] We not show the plots for the reparameterization and log-derivative trick, because we used a different computer for running these experiments, and the resulting plots would therefore not be meaningful. The computer was faster but both methods took significantly more time to converge compared to the natural gradient methods.

further expressing the gradient of the expectation parameters in terms of the gradient of the mean, $\nabla_{\boldsymbol{\mu}}$, and covariance, $\nabla_{\boldsymbol{\Sigma}}$, they derived the following update:

$$\left(\boldsymbol{\Sigma}^{(i+1)}\right)^{-1} = \left(\boldsymbol{\Sigma}^{(i)}\right)^{-1} - 2\beta_i \nabla_{\boldsymbol{\Sigma}} \mathcal{L}(q(\mathbf{x})), \qquad \boldsymbol{\mu}^{(i+1)} = \boldsymbol{\mu}^{(i)} + \beta_i \boldsymbol{\Sigma}^{(i+1)} \nabla_{\boldsymbol{\mu}} \mathcal{L}(q(\mathbf{x})), \tag{8}$$

where $\beta_i$ is the stepsize. Please refer to the original work for the derivation (Khan & Nielsen, 2018, Appendix C). As shown by Opper & Archambeau (2009), the gradient of an expected value $E_q\big[f(\boldsymbol{x})\big]$ with respect to mean and covariance can be expressed in terms of the gradient $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ and Hessian $\nabla_{\boldsymbol{xx}} f(\boldsymbol{x})$,

$$\nabla_{\boldsymbol{\mu}} E_q\big[f(\boldsymbol{x})\big] = E_q\big[\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\big], \qquad \nabla_{\boldsymbol{\Sigma}} E_q\big[f(\boldsymbol{x})\big] = \frac{1}{2} E_q\big[\nabla_{\boldsymbol{xx}} f(\boldsymbol{x})\big]. \tag{9}$$

We can directly use Equation 9 to estimate the natural gradient update (Eq. 8) of the ELBO objective (Eq. 1) based on a Monte-Carlo estimate of the expected gradient and Hessian of $f(\boldsymbol{x}) = \log \tilde{p}(\boldsymbol{x}) - \log q(\mathbf{x})$. However, as the gradient and Hessian of the Gaussian log density function are given by $\nabla_{\boldsymbol{x}} \log q(\mathbf{x}) = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{x})$ and $\nabla_{\boldsymbol{xx}} \log q(\mathbf{x}) = -\boldsymbol{\Sigma}^{-1}$, the natural gradient update for the ELBO simplifies to

$$\begin{aligned}
\left(\boldsymbol{\Sigma}^{(i+1)}\right)^{-1} =& (1-\beta_i)\left(\boldsymbol{\Sigma}^{(i)}\right)^{-1} - \beta_i E_q\big[\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x})\big] \\
\boldsymbol{\mu}^{(i+1)} =& \boldsymbol{\Sigma}^{(i+1)}\left(\boldsymbol{\Sigma}^{(i+1)}\right)^{-1}\left(\boldsymbol{\mu}^{(i)} + \beta_i \boldsymbol{\Sigma}^{(i+1)} \nabla_{\boldsymbol{\mu}\mathcal{L}(q(\mathbf{x}))}\right), \\
=& \boldsymbol{\Sigma}^{(i+1)}\left(\left((1-\beta_i)\left(\boldsymbol{\Sigma}^{(i)}\right)^{-1} - \beta_i E_q\big[\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x})\big]\right)\boldsymbol{\mu}^{(i)} + \beta_i \nabla_{\boldsymbol{\mu}} \mathcal{L}(q(\mathbf{x}))\right), \\
=& \boldsymbol{\Sigma}^{(i+1)}\left((1-\beta_i)\left(\boldsymbol{\Sigma}^{(i)}\right)^{-1}\boldsymbol{\mu}^{(i)} - \beta_i E_q\big[\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x})\big]\boldsymbol{\mu}^{(i)} + \beta_i E_q\big[\nabla_{\boldsymbol{x}} \log \tilde{p}(\boldsymbol{x})\big]\right),
\end{aligned} \tag{10}$$

that is, only the gradient and Hessian of the log target density need to be approximated with Monte-Carlo.

### A.5.1 VON

VON (Khan & Nielsen, 2018) further assumes that the target distribution has the typical form of a posterior distribution with a Gaussian prior $p_0(\boldsymbol{x}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$, that is,

$$\log \tilde{p}(\boldsymbol{x}) = \sum_{d \in \mathcal{D}} \log p(d|\boldsymbol{x}) + \log p_0(\boldsymbol{x}), \tag{11}$$

where $\log p(d|\boldsymbol{x})$ is the likelihood of a single training data point $d$. The Hessian of the log-density-function of the Gaussian prior distribution is constant and given by its negated inverse covariance matrix, and, thus, only the expected Hessian of the likelihood term needs to be estimated. An unbiased estimate of it can be obtained based on a minibatch of training data. VON uses a single sample for estimating the expected gradient and Hessian using Equation 9.

### A.5.2 VOGN

Computing the Hessian of the likelihood is often expensive, and might not even be possible. Hence, Khan & Nielsen (2018) proposed to apply the approximation used by the generalized Gauß-Newton method, which approximates the Hessian for each individual data point, by the outer product of the gradient. The resulting method, VOGN, hence estimates the Hessian of the ELBO as

$$E_q\big[\nabla_{\boldsymbol{xx}} f(\boldsymbol{x})\big] = E_q\big[\nabla_{\boldsymbol{xx}}\big[\log \tilde{p}(\boldsymbol{x}) - \log q(\mathbf{x})\big]\big] \approx \frac{|\mathcal{D}|}{|\mathcal{B}|} \sum_{d \in \mathcal{B}} \left[\boldsymbol{g}_d \boldsymbol{g}_d^\top\right] - \boldsymbol{\Sigma}_{\boldsymbol{p}} + \boldsymbol{\Sigma}^{(i)}, \tag{12}$$

where $\boldsymbol{g}_d = \nabla_{\boldsymbol{x}} \log(d|\boldsymbol{x})$ is the gradient of the likelihood of data point $d$, and $\mathcal{B} \subset \mathcal{D}$ is a minibatch.

### A.5.3 MORE

Comparing the update of MORE with $\omega = 1$ (Eq. 4) and the update given by Equation 10, we can observe that they share the same form and can be transformed to each other by substituting

- $\beta_i$ with $\frac{1}{\eta+1}$,
- $\boldsymbol{R}^{(i)}$ with $E_q\big[\nabla_{\boldsymbol{x}\boldsymbol{x}} \log \tilde{p}(\boldsymbol{x})\big]$,
- $\boldsymbol{r}^{(i)}$ with $E_q\big[\nabla_{\boldsymbol{x}\boldsymbol{x}} \log \tilde{p}(\boldsymbol{x})\big]\boldsymbol{\mu}^{(i)} - E_q\big[\nabla_{\boldsymbol{x}} \log \tilde{p}(\boldsymbol{x})\big]$ .

As also $\boldsymbol{R}^{(i)} = E_q\big[\nabla_{\boldsymbol{x}\boldsymbol{x}} R^{(i)}(\boldsymbol{x})\big]$ and $\boldsymbol{r}^{(i)} = E_q\big[\nabla_{\boldsymbol{x}\boldsymbol{x}} R^{(i)}(\boldsymbol{x})\big]\boldsymbol{\mu}^{(i)} - E_q\big[\nabla_{\boldsymbol{x}} R^{(i)}(\boldsymbol{x})\big]$ holds, we can see that MORE approximates the natural gradient with the exact natural gradient for a local Gaussian approximation of the target distribution, that is proportional to $exp\left(R^{(i)}(\boldsymbol{x})\right)$. Indeed, the quadratic surrogate is compatible with the Gaussian variational approximation in the sense stated by (Sutton et al., 2000), that is, the features of the surrogate, $\phi(\boldsymbol{x})$, correspond to the derivative of the log density function of the Gaussian (when using natural parameterization), and hence, its parameters correspond to an unbiased estimate of the natural gradient (Peters & Schaal, 2008; Pajarinen et al., 2019). The main difference between MORE, gMORE, VOGN, VON and GM is, thus, that they apply different methods to estimate the expected Hessian and gradient, which is crucial for the natural gradient update.

Apart from that, the methods only differ in the way they adapt the stepsize: VOGN, VON and GM apply a fixed schedule, whereas MORE and gMORE choose the stepsize such that the KL constraint $D_{\mathrm{KL}}(q^{(i+1)}(\boldsymbol{x})||q^{(i)}(\boldsymbol{x})) \leq \epsilon$ is satisfied.

### A.6 Details on the Implementation

Here we provide some details on our implementation of VIPS, MORE, VON, VOGN and GM. Our GVA optimizations were performed based on our VIPS implementation by setting the initial and maximum number of components to one (which leads to the GVA setting as the log-responsibilites become 0). We performed the evaluations of the reparameterization trick and the log-derivative trick based on the code published by Arenz et al. (2020).

#### A.6.1 VIPS

Our implementation of VIPS and MORE closely follows the C++ implementation published by Arenz et al. (2020). However, for some parts, we opted for a simpler implementation:

- Instead of performing a linesearch for initializing the covariance matrix of a newly added component, we always initialize them isotropic, which works similarly well, as already shown by Arenz et al. (2020).
- For reusing samples, we always reuse the $N_{\text{reuse}}$ newest samples; Arenz et al. (2020) use a heuristic to identify the most relevant among all previous samples.
- For optimizing the Lagrangian multiplier $\eta$, which relates to the stepsize during the component update, we use a line-search to identify the smallest value that satisfies the KL-bound, instead of optimizing the Lagrangian dual as proposed by Arenz et al. (2020). Our approach is sound due to the convexity of the dual function and was numerically more stable in our experiments.

Furthermore, instead of using the lower bound given by Equation 5, we use an equivalent formulation that was introduced by Becker et al. (2019). Namely, the lower bound can be reformulated to use a component specific reward $r_{o,\text{Becker}}(\boldsymbol{x}) = \log \tilde{p}(\boldsymbol{x}) - \log \tilde{q}(\boldsymbol{x})$ instead of $r_o(\boldsymbol{x}) = \log \tilde{p}(\boldsymbol{x}) + \log \tilde{q}(o|\boldsymbol{x})$ by replacing the entropy bonus of the component optimization by a KL-penalty:

$$
\begin{aligned}
L(q(\boldsymbol{x}), \tilde{q}(\boldsymbol{x})) &= \sum_o q(o) \Big[ \overbrace{\int_{\boldsymbol{x}} q(\boldsymbol{x}|o) \left(\log \tilde{p}(\boldsymbol{x}) + \log \tilde{q}(o|\boldsymbol{x})\right) d\boldsymbol{x} + H(q(\boldsymbol{x}|o))}^{J_o(q(\boldsymbol{x}|o))} \Big] + H(q(o)) \\
&= \sum_o q(o) \Big[ \underbrace{\int_{\boldsymbol{x}} q(\boldsymbol{x}|o) \left(\log \tilde{p}(\boldsymbol{x}) - \log \tilde{q}(\boldsymbol{x})\right) d\boldsymbol{x} - D_{\mathrm{KL}}(q(\boldsymbol{x}|o)||\tilde{q}(\boldsymbol{x}|o))}_{J_{o,\text{Becker}}(q(\boldsymbol{x}|o))} \Big] \\
&\quad - D_{\mathrm{KL}}(q(o)||\tilde{q}(o)),
\end{aligned}
$$

Table 2: Tested Hyperparameters per Experiment

| **Method** | $(N_{\textbf{desired}}, N_{\textbf{reused}})$ |
|---|---|
| gMORE | $\{(50, 20), (100, 50), (200, 100), (300, 150), (400, 200), (500, 250)\}$ |
| VON, VOGN, GM | $\{(0, 1), (5, 3), (20, 10), (50, 20), (100, 50), (200, 100)\}$ |

where $\tilde{q}(\boldsymbol{x})$ and $\tilde{q}(\boldsymbol{x}|o)$ are the auxiliary distributions that are set according to the approximation at the last iteration. The Becker-bound $J_{o,\text{Becker}}(q(\boldsymbol{x}|o))$ is more convenient to implement, and the additional term inside the KL, $\log \tilde{q}(\boldsymbol{x}|o)$ can be easily integrated into the natural gradient updates of MORE, VOGN, VON and GM, since it relates to an additional Gaussian prior for which the expected Hessian and gradient can be computed in closed form. Indeed, the only affect on our MORE implementation is that the Lagrangian multiplier $\eta$ is lower-bounded by $1$ rather than $0$, and that we use $\omega = 0$ instead of $\omega = 1$. Please note, that both formulations yield the same natural gradient update, except for negligible numerical errors.

### A.6.2 VON, VOGN AND GM

In our experiments, we use the same trust region updates that we use for MORE also when using VON, VOGN or GM, because we found the trust-region more robust in the GMM setting than the stepsize schedule used by Khan & Nielsen (2018). Please note that both approaches to adapt the stepsize are sound for all of the considered approaches and since their is a one-one-to mapping between $\eta$ and any $\beta < 1$ they are expected to perform similar for well-tuned hyperparameters. To compute the update, we set $\boldsymbol{R}^{(i)} = E_q\big[\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x})\big]$ and $\boldsymbol{r}^{(i)} = E_q\big[\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x})\big]\boldsymbol{\mu}^{(i)} - E_q\big[\nabla_{\boldsymbol{x}} \log \tilde{p}(\boldsymbol{x})\big]$ and proceed as if the surrogate was learned by MORE, by optimizing $\eta$ to satisfy a given KL-bound. Our implementation hence only differs in the way the expected Hessian and gradient for the natural gradient are computed.

- For VON, we have direct access to the the Hessian $\nabla_{\boldsymbol{xx}} r_{o,[Becker}$ and the gradient $\nabla_{\boldsymbol{x}} r_{o,\text{Becker}}$ and can, thus, use Monte-Carlo estimates to approximate the expectation under the current component,
- For VOGN, we estimate the Hessian for every individual training data point according to Equation 11, which is only possible for the logistic regression experiments. For the second part of our objective function $r_{o,[Becker} = \log \tilde{p}(\boldsymbol{x}) - \log \tilde{q}(\boldsymbol{x})$, we use Monte-Carlo estimates based on the analytical Hessian,
- For GM, we also make use of the analytical Hessian of $\log \tilde{q}(\boldsymbol{x})$, but do not assume the posterior structure (Eq. 11) and, hence, need to approximate the Hessian as $\nabla_{\boldsymbol{xx}} \log \tilde{p}(\boldsymbol{x}) \approx \big[\nabla_{\boldsymbol{x}} \log \tilde{p}\big]\big[\nabla_{\boldsymbol{x}} \log \tilde{p}\big]^\top$. Also for GM, we make use of the analytic form of the Hessian of the log-density of the GMM.

### A.6.3 HYPERPARAMETERS

We automatically adapt the KL bound based on the scheme proposed by Arenz et al. (2020), that is, we increase the trust region for a given component if its estimated performance on its objective $J_o(q(\boldsymbol{x}|o))$ improved, and decrease it otherwise. Hence, the only parameters that we need to tune relate to the samples. We follow the scheme proposed by Arenz et al. (2020) for drawing new samples. Namely, we specify the number of reused samples $N_{\text{reused}}$ and the number of desired samples $N_{\text{desired}}$. At every iteration we select the $N_{\text{components}} \times N_{\text{reuse}}$ newest samples for importance sampling. We then compute for every component the number of effective samples $n_{\text{eff}}(o)$ based on the importance weights, and draw $n_{\text{new}} = n_{\text{desired}} - \lfloor n_{\text{eff}}(o) \rfloor$ new samples.

We conducted a preliminary evaluation with a single seed for each algorithm (except for MORE, where we used the parameters from Arenz et al. (2020)) and test problem to select the $N_{\text{desired}}$ and $N_{\text{reused}}$. The candidates are given as tuples $(N_{\text{desired}}, N_{\text{reused}})$ in Table 2. We selected the hyperparameter for the GMM setting and used the same parameters for the respective GVA experiments.

The chosen hyperparameters per experiment are given in Table 3.

Table 3: Selected Hyperparameters per Experiment

| **Method** | **Breast Cancer** | **German Credit** | **Planar 1** | **Planar 4** | **GMM [20-100]** |
|---|---|---|---|---|---|
| gMORE | $(200, 100)$ | $(100, 50)$ | $(100, 50)$ | $(100, 50)$ | $(5N, 2.5N)$ |
| VON | $(20, 10)$ | $(5, 3)$ | $(5, 3)$ | $(5, 3)$ | $(100, 50)$ |
| VOGN | $(5, 3)$ | $(5, 3)$ | N/A | N/A | N/A |
| GM | $(100, 50)$ | $(100, 50)$ | $(100, 50)$ | $(100, 50)$ | N/A |

### A.6.4 Offsets for ELBO plots

To ensure positive values for our log-log plots of the ELBO (Fig 3 and Fig 1) to increase the resolution close to the optimal values, we subtracted constant offsets from the ELBOs, which were estimated based on two thousand samples. The offsets are given in Table 4.

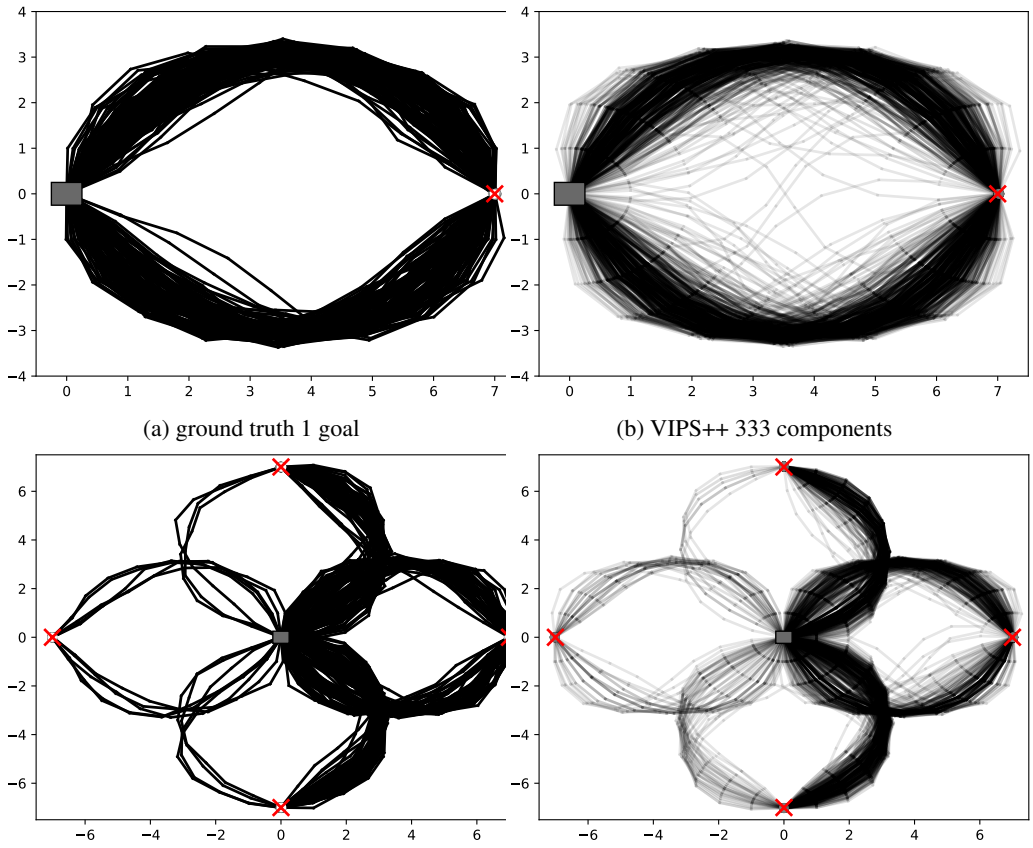
Table 4: Offsets subracted from the estimated ELBOs

| **Breast Cancer (BC)** | **German Credit (GC)** | **Planar 1** | **Planar 4** | **BC (GVA)** | **GC (GVA)** |
|---|---|---|---|---|---|
| 78.142 | 585.096 | 11.549 | 12.435 | 78.135 | 585.095 |

## A.7 Planar Robot Experiment

Here we visualize the weights and means learned in the planar robot experiments. Figure 7 shows plots of the ground-truth samples and of the components learned by Arenz et al. (2020).

Figure 8 shows the weights and means learned by different methods. As we can see from the plots, GM only find few modes, which indicates that many other components converged to bad configurations and where, thus, deleted. Interestingly, vonVIPS was unable to find all the modes. This is probably caused due to too little exploration by using fewer samples in each iteration. We did not detect this problem during our hyperparameter selection as the missing mode does not seem to affect the performance in terms of the ELBO. We believe, that vonVIPS should be able to find all modes, when using less aggressive hyperparameters (e.g., drawing more samples per iteration).



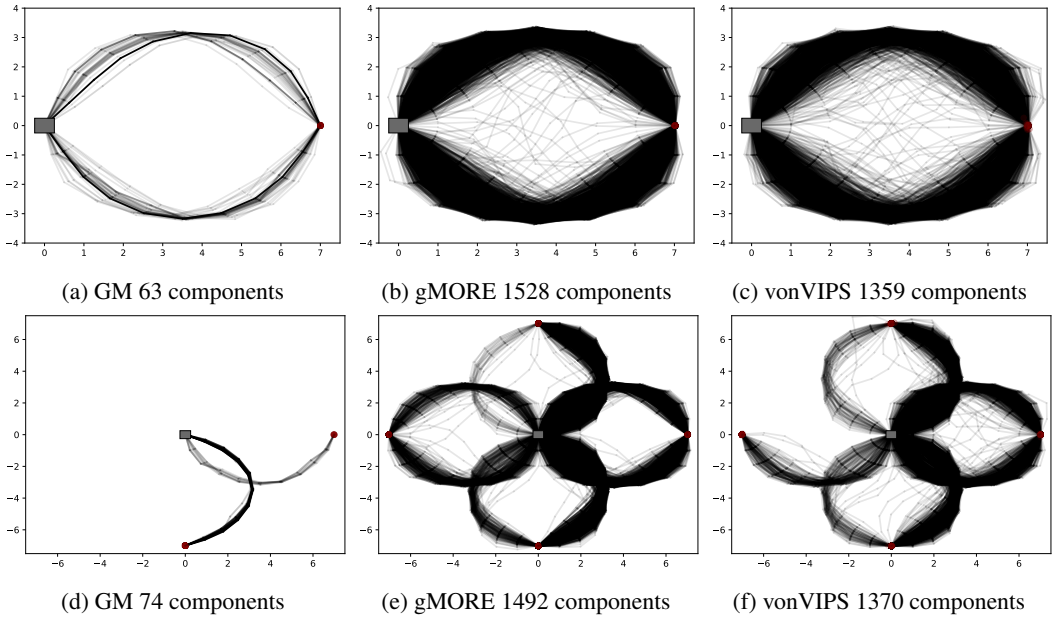
(a) ground truth 1 goal

(b) VIPS++ 333 components

(c) ground truth 4 goals

(d) VIPS++ 360 components

Figure 7: The left two plots show 200 ground-truth samples for both planar robot experiments. The right two plots visualize the weights and means of the mixture models learned by VIPS for each of the planar robot experiments. The ground-truth samples are generated using generalized elliptical slice sampling. The gray box indicates the base of the robot, the red crosses indicate the goal positions. Each line represents a component, components with larger weight are drawn darker. The planar robot 1 goal plot shows 333 components learned by VIPS and the planar robot 4 goals plot shows 360 components learned by VIPS. Pictures are taken from Arenz et al. (2020).



(a) GM 63 components (b) gMORE 1528 components (c) vonVIPS 1359 components

(d) GM 74 components (e) gMORE 1492 components (f) vonVIPS 1370 components

Figure 8: Visualization of learned means and weights in the planar robot one goal and four goals experiments. From left to right the plots are arranged as *GM* (left), *gMORE* (middle) and *vonVIPS* (right). The number of components differs from each other due to explore ability. Grey box indicates the robot base and red circle indicates the position of the end-effector. Each line represents a component, components with larger weight are drawn darker.