entropy

MDPI

Article

Flexible and Efficient Inference with Particles for the Variational Gaussian Approximation

Théo Galy-Fajou [1,*], Valerio Perrone [2] and Manfred Opper [1,3]

[1] Artificial Intelligence Group, Technische Universität Berlin, 10623 Berlin, Germany; manfred.opper@tu-berlin.de
[2] Amazon Web Services, 10969 Berlin, Germany; vperrone@amazon.com
[3] Centre for Systems Modelling and Quantitative Biomedicine, University of Birmingham, Birmingham B15 2TT, UK
* Correspondence: galy-fajou@tu-berlin.de

Abstract: Variational inference is a powerful framework, used to approximate intractable posteriors through variational distributions. The de facto standard is to rely on Gaussian variational families, which come with numerous advantages: they are easy to sample from, simple to parametrize, and many expectations are known in closed-form or readily computed by quadrature. In this paper, we view the Gaussian variational approximation problem through the lens of gradient flows. We introduce a flexible and efficient algorithm based on a linear flow leading to a particle-based approximation. We prove that, with a sufficient number of particles, our algorithm converges linearly to the exact solution for Gaussian targets, and a low-rank approximation otherwise. In addition to the theoretical analysis, we show, on a set of synthetic and real-world high-dimensional problems, that our algorithm outperforms existing methods with Gaussian targets while performing on a par with non-Gaussian targets.

Keywords: variational inference; Gaussian; particle flow; variable flow

Citation: Galy-Fajou, T.; Perrone, V.; Opper, M. Flexible and Efficient Inference with Particles for the Variational Gaussian Approximation. *Entropy* **2021**, *23*, 990. https://doi.org/10.3390/e23080990

Academic Editor: Pierre Alquier

Received: 22 June 2021
Accepted: 21 July 2021
Published: 30 July 2021

1. Introduction

Representing uncertainty is a ubiquitous problem in machine learning. Reliable uncertainties are key for decision making, especially in contexts where the trade-off between exploitation and exploration plays a central role, such as Bayesian optimization [1], active learning [2], and reinforcement learning [3]. While Bayesian inference is a principled tool to provide uncertainty estimation, computing posterior distributions is intractable for many problems of interest. Most sampling methods struggle to scale up to large datasets [4], while the diagnosis of convergence is not always straightforward [5]. On the other hand, *Variational Inference* **(VI)** methods can rely on well-understood optimization techniques and scale well to large datasets, at the cost of an approximation quality depending heavily on the assumptions made. The Gaussian family is by far the most popular variational approximation used in VI [6,7]. This is for several reasons. First, Gaussian variational families are easy to sample from, reparametrize, and marginalize. Second, they are easily amenable to diagonal covariance approximations, making them scalable to high dimensions. Third, most expectations are either easily computable by quadrature or Monte Carlo integration, or known in closed-form.

A large body of work covers different approaches to optimize the *Variational Gaussian Approximation* **(VGA)**, with the speed of convergence and the scalability in dimensions as the main concerns. From the perspective of convergence speed, the major bottleneck when computing gradients with stochastic estimators is the estimator variance [8]. *Particle-based methods* with deterministic paths do not have this issue, and have been proven to be highly successful in many applications [9–11]. However, can we use a particle-based

algorithm to compute a VGA? If so, what are its properties and is it competitive with other VGA methods?

In this paper, we attempt to answer these questions by introducing the *Gaussian Particle Flow* **(GPF)**, a framework to approximate a Gaussian variational distribution with particles. GPF is derived from a continuous-time flow, where the necessary expectations over the evolving densities are approximated by particles. The complexity of the method grows quadratically with the number of particles but linearly with the dimension, remaining compatible with other approximations such as structured mean-field approximations. Using the same dynamics, we also derive a stochastic version of the algorithm, *Gaussian Flow* **(GF)**. To show convergence, we prove the decrease in an empirical version of the free energy that is valid for a finite number of particles. For the special case of D–dimensional Gaussian target densities, we show that $D+1$ particles are enough to obtain convergence to the true distribution. We also find, for this case, that convergence is exponentially fast. Finally, we compare our approach with other VGA algorithms, both in fully controlled synthetic settings and on a set of real-world problems.

2. Related Work

The goal of Bayesian inference is to carry out computations with the posterior distribution of a latent variable $x \in R^D$ given some observations y. By Bayes theorem, the posterior distribution is $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$, where $p(y|x)$ and $p(x)$ are, respectively, the likelihood and the prior distribution. Even if the likelihood and the prior are known analytically, marginalizing out high-dimensional variables in the product $p(y|x)p(x)$ in order to compute quantities such as $p(y)$ is typically intractable. *Variational Inference* **(VI)** aims to simplify this problem by turning it into an optimization one. The intractable posterior is approximated by the closest distribution within a tractable family, with closeness being measured by the *Kullback-Leibler* **(KL)** divergence, defined by

$$\mathrm{KL}\left[q(x)||p(x)\right] = \mathbb{E}_q[\log q(x) - \log p(x)],$$

where $\mathbb{E}_q[f(x)] = \int f(x)q(x)dx$ denotes the expectation of f over q. Denoting by $\mathcal{Q}$ a family of distributions, we look for

$$\arg\min_{q\in\mathcal{Q}} \mathrm{KL}\left[q(x)||p(x|y)\right].$$

Since $p(y)$ is not computable in an efficient way, we equivalently minimize the upper bound $\mathcal{F}$:

$$\mathrm{KL}[q(x)||p(x|y)] \leq \mathcal{F}[q] = -\mathbb{E}_q[\log p(y|x)p(x)] - \mathbb{H}_q, \tag{1}$$

where $\mathbb{H}_q$ is the entropy of q ($-\mathbb{E}_q[\log q(x)]$). Here, $\mathcal{F}$ is known as the variational free energy and $-\mathcal{F}$ is known as the Evidence Lower BOund (ELBO). A diverse set of approaches to perform VI with Gaussian families $\mathcal{Q}$ have been developed in the literature, which we review in the following.

2.1. The Variational Gaussian Approximation

The VGA is the restriction of $\mathcal{Q}$ to be the family of multivariate Gaussian distributions $q(x) = \mathcal{N}(m, C)$, where $m \in \mathbb{R}^D$ is the mean and $C \in \{A \in \mathbb{R}^{D\times D} | x^\top Ax \geq 0, \forall x \in \mathbb{R}^D\}$ is the covariance matrix, for which the free energy is found to be

$$\mathcal{F}[q] = -\frac{1}{2}\log|C| + \mathbb{E}_q[\varphi(x)]. \tag{2}$$

where $\varphi(x) = -\log(p(y|x)p(x))$. A standard descent algorithm based on gradients of Equation (2) with respect to variational parameters m, C give rise to some issues. First, naively computing the gradient of the expectation with respect to the covariance matrix

C involves unwanted second derivatives of $\varphi(x)$ [12], which may not be available or may be computationally too expensive in a *black-box* setting. Second, the gradient of the entropy term $\mathbb{H}_q$ entails inverting a non-sparse matrix, which we would like to avoid for higher-dimensional cases. Finally, the positive-definiteness of the covariance matrix leads to non-trivial constraints on parameter updates, which can lead to a slowdown of convergence or, if ignored, to instabilities in the algorithm.

To solve these issues, a variety of approaches have been proposed in the literature. If we focus on factorizable models, we can make a simplification: for problems with likelihoods that can be rewritten as $p(y|x) = \prod_{d=1}^{D} p(y|x_d)$, the number of independent variational parameters is reduced to $2D$ [12,13]. In this special case, the Gaussian expectations in the free energy (2) split into a sum of 1-dimensional integrals, which can be efficiently computed by using numerical quadrature methods. To extend to the general case, gradients of the free energy are estimated by a stochastic sampling approach, which also forms the starting point of our method. This relies on the so-called *reparametrization trick*, where the expectation over the parameter-dependent variational density q_θ is replaced by an expectation over a fixed density q^0 instead. This facilitates the gradient computation because unwanted derivatives of the type $\nabla_\theta q_\theta(x)$ are avoided. For the Gaussian case, the reparametrization trick is a linear transformation of an arbitrary D dimensional Gaussian random variable $x \sim q_\theta(x)$ in terms of a D-dimensional Gaussian random variable $x^0 \sim q^0 = \mathcal{N}(m^0, C^0)$:

$$x = \Gamma(x^0 - m^0) + m, \tag{3}$$

where $\Gamma \in \mathbb{R}^{D\times D}$ and $m \in \mathbb{R}^D$ are the variational parameters. We assume that the covariance C^0 is not degenerate and, for simplicity, we set it as the identity. For instance, the gradient of the expectation given q over a function f given the mean m becomes $\nabla_m \mathbb{E}_q[f(x)] = \mathbb{E}_{q^0}\left[\nabla_m f(\Gamma(x^0 - m^0) + m)\right]$. This can be simply proved by using the reparametrization (3) inside the integral and passing the gradient inside; for more details, see [14].

Given this representation, the free energy is easily obtained as a function of the variational parameters:

$$\mathcal{F}(q) = -\log|\Gamma| + \mathbb{E}_{q^0}\Big[\varphi(\Gamma(x^0 - m^0) + m)\Big]. \tag{4}$$

Other representations are possible. Challis and Barber [13] and Ong et al. [15] use a different reparametrization with a factorized structure of the covariance $C = \Gamma^\top\Gamma + \mathrm{diag}(d)$, where $\Gamma \in \mathbb{R}^{D\times P}$ and $d \in \mathbb{R}^D$, with $P \leq D$ is the rank of $\Gamma^\top\Gamma$. Other representations assume special structures of the precision matrix $\Lambda = C^{-1}$, which allow you to enforce special properties, such as sparsity in [16,17].

In general, these methods tend to scale poorly with the number of dimensions, as one needs to optimize $D(D+3)/2$ parameters. The (structured) *Mean-Field* **(MF)** [18,19] approach imposes independence between variables in the variational distribution. The number of variational parameters is then $2D$, but covariance information between dimensions is lost.

2.2. Natural Gradients

Besides the issue of expectations, more efficient optimizations directions, beyond ordinary gradient descent, have been considered. These can help to deal with constraints such as those given for the covariance matrix. Natural gradients [20] are a special case of Riemannian gradients and utilize the specific Riemannian manifold structure of variational parameters. They can often deal with constraints of parameters (such as the positive definiteness of the covariance), accelerate inference, and improve the convergence of algorithms. The application of such advanced gradient methods typically requires an estimate of the inverse Fisher information matrix as a preconditioner of ordinary gradients. Khan and Nielsen [21] and Lin et al. [22] propose a solution that requires extra second derivatives of the log–posteriors. Salimbeni et al. [23] developed an automatic process to

compute these without the second derivatives but with instability issues. Lin et al. [17] solved these issues by using geodesics on the manifold of parameters, at the price of having to compute inverse matrices as well as Hessians.

2.3. Particle-Based VI

Stochastic gradient descent methods compute expectations (and gradients) at each time step with new independent Monte Carlo samples drawn from the current approximation of the variational density. Particle-based methods for variational inference *draw samples only once* at the beginning of the algorithm instead. They iteratively construct transformations of an initial random variable (having a simple tractable density) where the transformed density leads to the decrease and finally to the minimum of the variational free energy. The iterative approach induces a deterministic temporal flow of random variables which depends on the current density of the variable itself. Using an approximation by the empirical density (which is represented by the positions of a set of 'particles') one obtains a flow of interacting particles which converges asymptotically to an empirical approximation of the desired optimal variational density.

The most popular approach is *Stein Variational Gradient Descent* **(SVGD)** [24], which computes a nonparametric transformation based on the kernelized Stein discrepancy [9]. SVGD has the advantage of not being restricted to a parametric form of the variational distribution. However, using standard distance-based kernels like the squared exponential kernel ($k(x, y) = \exp(-\|x - y\|_2^2/2)$) can lead to underestimated covariances and poor performance in high dimensions [11,25]. Hence, it is interesting to develop particle approaches that approximate the VGA. We provide a more thorough comparison between our method and SVGD in Section 3.6.

2.4. GVA in Bayesian Neural Networks

There has been increased interest in making *Bayesian Neural Networks* **(BNN)** by adding priors to Neural Networks parameters. The true form of the posterior is unknown but VGA has been used due to its ease of use and scalability with the number of dimensions (typically $D \gg 10^5$). Most of the aforementioned methods apply to BNN, but techniques have been specifically tailored with BNN in mind. [26] use the low-rank structure of [13] but exploit the *Local Reparametrization Trick*, where each datapoint y_i gets a different sample from q in order to reduce the stochastic gradient estimator variance. *Stochastic Weight Averaging-Gaussian* **(SWAG)** [27], in which a set of particles obtained via stochastic gradient descent represent a low-rank Gaussian distribution, approximating the true posterior with a prior posterior produced by the network's regularization. While easy to implement, SWAG does not allow you to incorporate an explicit prior, and the resulting distribution does not derive from a principled Bayesian approach.

2.5. Related Approaches

The closest approach to our proposed method is the *Ensemble Kalman Filter* **(EKF)** [28]. It assumes that the posterior is computed in a sequential way, where, at each time step, only single (or smaller batches) of data observations, represented by their likelihoods, become available. An ensemble of particles, representing a Gaussian distribution is iteratively updated with every new batch of observations. EKF allows us to work on high-dimensional problems with a limited amount of particles but is restricted to factorizable likelihoods for which a sequential representation is possible. While EKF maintains a representation of a Gaussian posterior, it is not clear how this relates to the goal of minimizing the free energy or the KL divergence.

3. Gaussian (Particle) Flow

We introduce *Gaussian Particle Flow* **(GPF)** and *Gaussian Flow* **(GF)**, two computationally tractable approaches, to obtain a *Variational Gaussian Approximation* **(VGA)**. In the following, we derive deterministic linear dynamics, which decreases the variational free

energy. We additionally give some variants with a *Mean-Field* **(MF)** approach and prove theoretical convergence guarantees.

In the following, $\frac{d(\cdot)}{dt}$ indicates the total derivative given time, $\frac{\partial(\cdot)}{\partial t}$ partial derivatives given time, $\nabla_x(\cdot)$ gradients given a vector x.

3.1. Gaussian Variable Flows

We next discuss an alternative approach to generate the desired transformation of random variables, leading from a simple (prior) Gaussian density to a more complex Gaussian, which minimizes the variational free energy. It is based on the idea of *variable flows*, i.e., recursive deterministic transformations of the random variables defined by a mapping $x^{n+1} = x^n + \epsilon f^n(x^n)$ where $f^n : \mathbb{R}^D \to \mathbb{R}^D$. Well-known examples of flows are *Normalizing Flows* [29], where f^n are bijections, or *Neural ODEs* [30] where $f^n = f$ is defined by a neural network and x^0 is the input. For simplicity, we will consider small changes $\epsilon \to 0$ and work with flows in the continuous-time limit ($t = n\epsilon$), which follow a system of *Ordinary Differential Equation* **(ODE)**. For the Gaussian case, in the spirit of the reparametrization trick (3), we choose a linear corresponding map f and write

$$\frac{dx^t}{dt} = f^t(x^t) = A^t(x^t - m^t) + b^t, \tag{5}$$

where A^t is a matrix and $m^t \doteq \mathbb{E}_{q^t}[x]$ (which is no longer interpreted as an independent variational parameter). When the initial random variable x^0 is Gaussian distributed, the vectors x^t are also Gaussian for any t. To construct a flow that decreases the free energy over time, we can either compute the time derivative of the specific free energy (2) induced by the ODE (5), or simply derive the general result valid for smooth maps f (see, e.g., [24]). To be self contained, we briefly repeat the main steps: We first compute the change of the free energy in terms of the time derivative of q^t:

$$\begin{aligned}
\frac{d\mathcal{F}[q^t]}{dt} &= \frac{d}{dt}\int q^t(x)\big(\log q^t(x) + \varphi(x)\big)dx \\
&= \int \frac{\partial q^t(x)}{\partial t}\big(\log q^t(x) + \varphi(x)\big)dx + \int q^t(x)\left(\frac{\partial q^t(x)}{\partial t}\frac{1}{q^t(x)} + \frac{\partial \varphi(x)}{\partial t}\right)dx \\
&= \int \frac{\partial q^t(x)}{\partial t}\big(\log q^t(x) + \varphi(x)\big)dx
\end{aligned}$$

where we have used the fact that $\int \frac{\partial q^t(x)}{\partial t}dx = \frac{d}{dt}\int q^t(x)dx = 0$ and $\frac{\partial \varphi(x)}{\partial t} = 0$. We next use the *continuity equation* for the density

$$\frac{\partial q^t(x)}{\partial t} = -\nabla_x \cdot \big(q^t(x) f^t(x)\big),$$

related to the deterministic flow to obtain

$$\begin{aligned}
\frac{d\mathcal{F}[q^t]}{dt} &= \int \nabla_x \cdot \big(q^t(x) f^t(x)\big)\big(\log q^t(x) + \varphi(x)\big)dx \\
&= -\int \big(q^t(x) f^t(x)\big) \cdot \nabla_x \big(\log q^t(x) + \varphi(x)\big)dx \\
&= \int \big(\nabla_x \cdot (q^t(x) f^t(x)) + q^t(x) f^t(x) \cdot \nabla_x \varphi(x)\big)dx \\
&= \int \nabla_x q^t(x) \cdot f^t(x) + q^t(x) f^t(x) \cdot \nabla_x \varphi(x) dx \\
&= -\mathbb{E}_{q^t}\left[\nabla_x \cdot f^t(x) - f^t(x) \cdot \nabla_x \varphi(x)\right]
\end{aligned}$$

where we have applied Green's identity twice and used the fact that $\lim_{x\to\infty} q_t(x) = 0$. Specializing to the linear flow (5), we obtain

$$\frac{d\mathcal{F}[q^t]}{dt} = -\mathrm{tr}[A^t(A^t_\star)^\top] - (b^t)^\top b^t_\star, \tag{6}$$

where

$$\begin{aligned} A^t_\star \doteq & I - \mathbb{E}_{q^t}\left[\nabla_x \varphi(x)(x-m^t)^\top\right] \\ b^t_\star \doteq & -\mathbb{E}_{q^t}[\nabla_x \varphi(x)] \end{aligned} \tag{7}$$

Equation (6) represents the change in the free energy $\mathcal{F}$ for an infinitesimal change in the variables x given by the flow (5). Obviously, the simplest choices

$$A^t \equiv A^t_\star \qquad b^t \equiv b^t_\star \tag{8}$$

lead to a decrease in the free energy $\frac{d\mathcal{F}[q^t]}{dt} \leq 0$. More detailed derivations are given in Appendix A. Additionally, *equality* only happens, when

$$\begin{aligned} I - \mathbb{E}_q\left[\nabla_x \varphi(x)(x-m)^\top\right] &= 0 \\ \mathbb{E}_q[\nabla_x \varphi(x)] &= 0 \end{aligned} \tag{9}$$

Using Stein's lemma [31], we can show that these fixed-point solutions are equal to the conditions for the optimal variational Gaussian distribution solution given in [12]. In Appendix C, we show that our parameter updates can be interpreted as a Riemannian gradient descent method for the free energy (4). This is based on the metric introduced by ([20], Theorem 7.6) as an efficient technique for learning the mixing matrix in models of blind source separation. This gradient should not be confused with the so-called *natural gradient* obtained by pre-multiplying with the inverse Fischer-information matrix.

Of course, there are other choices for A^t and b^t, which lead to a decrease in the free energy and the same fixed-point equations. In Section 3.6, we discuss how SVGD, with a linear kernel, can lead to the same fixed points but with different dynamics.

3.2. From Variable Flows to Parameter Flows

Before we introduce the particle algorithm, we show that the results for the variable flow can also be converted into a temporal change of the parameters Γ^t, m^t, as defined for Equation (3). From this, a corresponding *Gaussian Flow* **(GF)** algorithm can be easily derived. By differentiating the parametrisation $x^t = \Gamma^t(x^0 - m^0) + m^t$ (with m^t now considered as free variational parameter) with respect to time t and using (5), we obtain

$$\frac{dx^t}{dt} = \frac{d\Gamma^t}{dt}(x^0 - m^0) + \frac{dm^t}{dt} = A^t(x^t - m^t) + b^t \tag{10}$$

By inserting $x^t = \Gamma^t(x^0 - m^0) + m^t$ into the right hand side of (10), and using the optimal parameters from (7), we obtain

$$\begin{aligned} \frac{d\Gamma^t}{dt} =& \Gamma^t - \mathbb{E}_{q^0}\left[\nabla_x \varphi(x^t)(x^0 - m^0)^\top\right]\Gamma^t(\Gamma^t)^\top \\ \frac{dm^t}{dt} =& -\mathbb{E}_{q^0}\left[\nabla_x \varphi(x^t)\right] \end{aligned} \tag{11}$$

Note that the expectations are over the probability distribution of the initial random variable x^0. Discretizing Equations (11) in time, and estimating the expectations by drawing independent samples from the fixed Gaussian q^0 at each time step, we obtain our GF algorithm to minimize the variational free energy in the space of Gaussian densities. We summarize the steps of GF in Algorithm 1. Remarkably, this scheme differs from previous VGA algorithms with Riemannian gradients based on the Fisher information

metric (see, e.g., [17,32]) because no *matrix inversions* or *second order derivatives* of the function φ are required.

GF also allows for the computation of a low-rank VGA by enforcing $\Gamma \in \mathbb{R}^{D\times K}$ and $x^0 \in \mathbb{R}^K$. This algorithm scales linearly in the number of dimensions and quadratically in the rank K of the covariance.

It is interesting to note that the reverse construction of a variable flow from a parameter flow is, in general, not possible. This would require the ability to eliminate all variational parameters and the initial variables x^0 in the resulting differential equation for x^t, and replace them with functions of x^t alone. For instance, if we eliminate the initial variables x^0 in terms of $(\Gamma^t)^{-1}$ and x^t the algorithm of [14], the resulting expression still depends on Γ^t.

3.3. Particle Dynamics

The main idea of the particle approach is to approximate the Gaussian density q^t in (7) by the empirical distribution

$$\hat{q}^t \doteq \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i^t) \tag{12}$$

computed from N samples x_i^t, $i = 1,\dots,N$. These are initially sampled from the density q^0 at time $t = 0$ and are then propagated using the discretized dynamics of the ODE (5):

$$\frac{dx_i^t}{dt} = -\eta_1^t \mathbb{E}_{\hat{q}^t}[\nabla_x \varphi(x)] - \eta_2^t \hat{A}^t (x_i^t - \hat{m}^t) \tag{13}$$

where

$$\hat{A}^t = I - \frac{1}{N}\sum_{i=1}^{N}\nabla_x\varphi(x)(x_i^t - \hat{m}^t)^\top$$

$$\hat{b}^t = \frac{1}{N}\sum_{i=1}^{N}\nabla_x\varphi(x_i^t), \qquad \hat{m}^t = \frac{1}{N}\sum_{i=1}^{N}x_i^t$$

where η_1^t and η_2^t are learning rates (We further comment on the use of different optimization schemes in Section 4.4). Note that although $\mathbb{E}_{\hat{q}^t}\left[\nabla_x\varphi(x)(x-\hat{m}^t)^\top\right]$ is a $D \times D$ matrix, changing the matrix multiplication order leads to a computational complexity of $\mathcal{O}(N^2D)$ with a storage complexity of $\mathcal{O}(N(N+D))$, since neither the empirical covariance matrix or A^t need to be explicitly computed.

Relaxation of Empirical Free Energy and Convergence

We have shown that the continuous-time dynamics (10) of the random variables leads to a decay of the free energy $\mathcal{F}(q^t)$ with time t. Assuming that the free energy is bounded from below, one might conjecture that this property would imply the convergence of the particle algorithm to a fixed point when learning rates are sufficiently small such that the discrete-time dynamics are approximated well by the continuous limit. Unfortunately, the finite number N of particles poses an extra problem. The definition of the free energy $\mathcal{F}(q)$ by the KL–divergence (1) for continuous random variables such as assumes that both $q(\cdot)$ and $p(\cdot|y)$ are densities with respect to the Lebesgue measure. Hence, $\mathcal{F}(\hat{q})$ is not defined if we take $q \equiv \hat{q}$, (12) as the empirical distribution of the finite particle approximation. Nevertheless, we define a finite N *approximation* to the Gaussian free energy, which is also then found to decay under the finite N dynamics. Let us first assume that $N > D$ and define

$$\tilde{\mathcal{F}}(\hat{q}^t) \doteq -\frac{1}{2}\log|\hat{C}^t| + \mathbb{E}_{\hat{q}^t}[\varphi(x)] \tag{14}$$

with the empirical covariance matrix

$$\hat{C}^t = \frac{1}{N}\sum_{i=1}^{N}\left(x_i^t - m^t\right)\left(x_i^t - m^t\right)^\top \tag{15}$$

The definition (14) is chosen in such way that in the large N limit, when the empirical distribution $\hat{q}^t$ converges to a Gaussian distribution q^t, we will also obtain the convergence of the approximation (14) to $\mathcal{F}(q^t)$. It can be shown (see Appendix B) that $\frac{d\tilde{\mathcal{F}}(\hat{q}^t)}{dt} \leq 0$, with equality only at the fixed points of the dynamics.

In applications of our particle method to high-dimensional problems, the limitations of computational power may force us to restrict particle numbers to be smaller than the dimensionality D. For $N < D+1$, the empirical covariance C^t will be singular, and typically contain only $N-1$ non-zero eigenvalues, which leads to the $-\log|\hat{C}| = \infty$ and makes Equation (14) meaningless. We resolve this issue through a regularisation of the log–determinant term in (14), replacing all zero eigenvalues of $\hat{C}$ by the values 1, i.e., $\lambda_i = 0 \to \tilde{\lambda}_i = 1$. We show in Appendix B that the free energy still decays, provided that the dynamics of the particles stay the same. This regularisation step can be formally stated as a replacement of the empirical covariance (15) in (14) by

$$\hat{C}^t \to \hat{C}^t + \sum_{i:\lambda_i^t=0} e_i^t (e_i^t)^\top$$

where $e_i^t = i$th eigenvector of $\hat{C}^t$.

3.4. Algorithm and Properties

The algorithm we propose is to sample N particles $\{x_1^0, \ldots, x_N^0\}$ where $x_i^0 \in \mathbb{R}^D$ from q^0 (which can be centered around the MAP for example), and iteratively optimize their positions using Equation (13). Once convergence is reached, i.e., $\frac{d\mathcal{F}}{dt} = 0$, we can easily make predictions using the converged empirical distribution $\hat{q}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i)$, where δ is the Dirac delta function, or, alternatively, the Gaussian density it represents, i.e., $q(x) = \mathcal{N}(m, C)$, where $m = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $C = \frac{1}{N}\sum_{i=1}^{N}(x_i - m)(x_i - m)^\top$. To draw samples from $\hat{q}$, no inversions of the empirical covariance C are needed, as we can obtain new samples by computing:

$$x = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}(x_i - m)\circ\xi_i + m, \tag{16}$$

where ξ_i are i.i.d. normal variables: $\xi_i \sim \mathcal{N}(0, \mathbb{I}_D)$. This can be shown by defining D, the deviation matrix, a matrix which columns equal to $D_i = \frac{x_i - m}{\sqrt{N}}$. We naturally have $DD^\top = C$ which makes D the Cholesky decomposition of C.

All the inference steps are summarized in Algorithm 2 and an illustration in two dimensions is provided in Figure 1.

We summarize the principal points of our approach:

- Gradients of expectations have zero variance, at the cost of a bias decreasing with the number of particles and equal to zero for Gaussian target (see Theorem 1);
- It works with noisy gradients (when using subsampling data, for example);
- The rank of the approximated covariance C is $\min(N-1, D)$. When $N \leq D$, the algorithm can be used to obtain a low-rank approximation.
- The complexity of our algorithm is $\mathcal{O}(N^2 D)$ and storing complexity is $\mathcal{O}(N(N+D))$. By adjusting the number of particles used, we can control the performance trade-off;
- GPF (and GF) are also compatible with any kind of structured MF (see Section 3.5);
- Despite working with an empirical distribution ,we can compute a surrogate of the free energy $\mathcal{F}(q)$ to optimize hyper-parameters, compute the lower bound of the log-evidence, or simply monitor convergence.

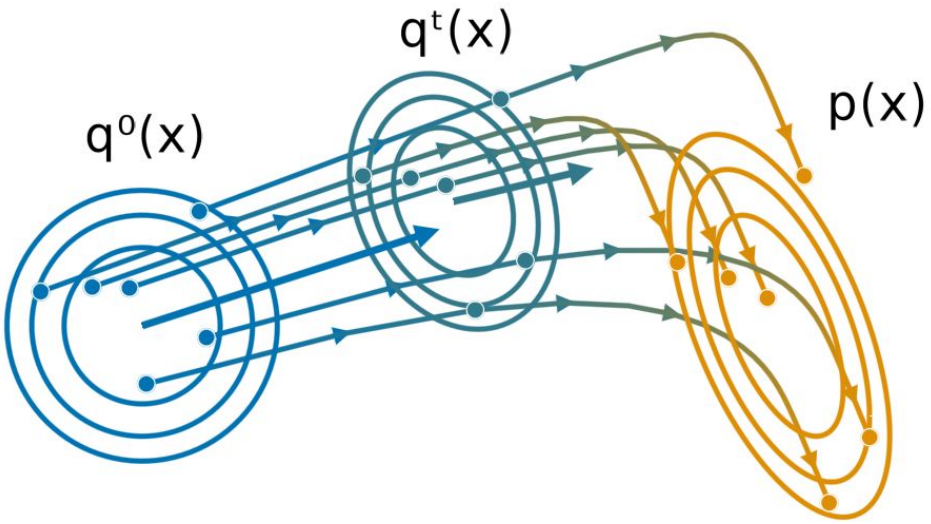


Figure 1. Illustration of the Gaussian Particle Flow algorithm, with $q^0(x)$ and $p(x)$ representing the initial and target distribution respectively. Particles are iteratively moved according to the gradient flow starting from $q^0(x)$, approximating a new Gaussian distribution $q^t(x)$ at each iteration t.

Algorithm 1: Gaussian Flow (GF)

Input: Number of samples N, initial distribution $q^0 = \mathcal{N}(\mu^0, \Gamma^0(\Gamma^0)^\top)$, target $p(x) \propto e^{-\varphi(x)}$, learning rates η_1^t, η_2^t
Output: Variational dist. $q(x) = \mathcal{N}(\mu, \Gamma\Gamma^\top)$
for t *in* $0 : T$ **do**

$\{x_i^0\}_{i=1}^N \sim q^0$ `# Sample N initial particles from q^0`
$x_i = \Gamma^t(x_i^0 - \mu^0) + \mu^t, \ \forall i$ `# Reparametrize`
$g_i = \nabla_x \varphi(x_i), \ \forall i$ `# Compute gradients`
$\mu^{t+1} = \mu^t - \eta_1^t \frac{1}{N} \sum_{i=1}^N \varphi(x_i)$ `# Update μ`
$A = \frac{1}{N} \sum_i g_i (x_i^0 - \mu^0)^\top (\Gamma^t)^\top$ `# Compute matrix`
$\Gamma^{t+1} = \Gamma^t - \eta_2^t A \Gamma^t$ `# Update Γ`

Algorithm 2: Gaussian Particle Flow (GPF)

Input: Number of particles N, initial distribution q^0, target $p(x) \propto e^{-\varphi(x)}$, learning rates η_1^t, η_2^t
Output: Empirical dist. $q(x) = \frac{1}{N} \sum_{i=1}^N \delta_{x,x_i}$
Init: *Sample N particles from* q^0 *:* $\{x_i^0\}_{i=1}^N$
for t *in* $0 : T$ **do**

$g_i = \nabla_x \varphi(x_i^t), \ \forall i$ `# Compute gradients`
$m = \frac{1}{N} \sum_i x_i, \quad \overline{g} = \frac{1}{N} \sum_i g_i$ `# Compute means`
$A = \frac{1}{N} \sum_i g_i (x_i^t - m)^\top - I$ `# Compute matrix`
$x_i^{t+1} = x_i^t - \eta_1^t \overline{g} - \eta_2^t A(x_i^t - m), \ \forall i$ `# Update particles`

3.4.1. Relaxation of Empirical Free Energy

The definition of the free energy $\mathcal{F}(q)$ from the KL–divergence (1) for a continuous random variables assumes that both $q(\cdot)$ and $p(\cdot|y)$ are densities with respect to the Lebesgue measure. Hence, it is not *a priori* clear that a specific *approximation* $\mathcal{F}(\hat{q}^t)$, based on an empirical distribution $\hat{q}^t(x) \doteq \frac{1}{N} \sum_{i=1}^N \delta(x - x_i^t)$ with a *finite number of particles* N, will decrease under the particle flow. Thus we may not be able to guarantee convergence to a fixed point for finite N. Luckily, as we show in Appendix D, we find that:

$$\frac{d\mathcal{F}(\hat{q}_t)}{dt} = \frac{d(\mathbb{E}_{\hat{q}_t}[\varphi(x)] - \frac{1}{2}\log|C^t|)}{dt} \leq 0. \tag{17}$$

For $N < D+1$, the empirical covariance C^t will typically contain $N-1$ non-zero eigenvalues and lead to $-\log|C| = \infty$, making Equation (17) meaningless. We resolve this issue by introducing a *regularized free energy* $\widetilde{\mathcal{F}}$ where $\log|C^t|$ is replaced by $\sum_{i:\lambda_i>0}\log\lambda_i$ where $\{\lambda_i\}_{i=1}^D$ are the eigenvalues of C^t. We show in Appendix D that, given the dynamics from Equation (5), $\widetilde{\mathcal{F}}$ is also guaranteed to not increase over time. It can, therefore, be used as a regularized proxy for the true $\mathcal{F}$ and used to optimize over hyper-parameters or to monitor convergence. Note that similar proofs exist for SVGD [33] and were proven to be highly non-trivial.

3.4.2. Dynamics and Fixed Points for Gaussian Targets

We illustrate our method by some exact theoretical results for the dynamics and the fixed points of our algorithm when *the target is a multivariate Gaussian density*. While such targets may seem like a trivial application, our analysis could still provide some insight into the performance for more complicated densities.

Theorem 1. *If the target density $p(x)$ is a D-dimensional multivariate Gaussian, only $D+1$ particles are needed for Algorithm 2 to converge to the exact target parameters.*

Proof. The proof is given in Appendix E. □

Theorem 2. *For a target $p(x) = \mathcal{N}(x \mid \mu, \Lambda^{-1})$, i.e., with precision matrixΛ, where $x \in \mathbb{R}^D$, and $N \geq D+1$ particles, the continuous time limit of Algorithm 2 will converge exponentially fast for both the mean and the trace of the precision matrix:*

$$\begin{aligned} m^t - \mu &= e^{-\Lambda t}(m^0 - \mu), \\ \mathrm{tr}((C^t)^{-1} - \Lambda) &= e^{-2t}\mathrm{tr}(\left(C^0\right)^{-1} - \Lambda), \end{aligned}$$

where m^t and C^t are the empirical mean and covariance matrix at time t and $\exp(-\Lambda t)$ is the matrix exponential.

Proof. The proof is given in Appendix F. □

Our result shows that convergence of the mean m^t directly depends on Λ. However, we can also precondition the gradient on m by C^t, i.e., using the natural gradient approximation in the Fisher sense, and eventually get rid of the dependency on Λ when $(C^t)^{-1} \approx \Lambda$.

The exponential relaxation of fluctuations also manifests itself in the decay of the free energy towards its minimum. For the Gaussian target, the free energy exactly separates into two terms corresponding to the mean and fluctuations. We can write $\mathcal{F}(m^t, C^t) = \frac{1}{2}(m^t-\mu)^\top\Lambda(m^t-\mu) + \frac{D}{2} + \mathcal{F}_{fl}(C^t)$, where the nontrivial fluctuation part (subtracted by its minimum) is given by

$$\mathcal{F}_{fl}(C^t) = -\frac{1}{2}\log|C^t| + \frac{1}{2}\mathrm{tr}(\Lambda C^t - I).$$

We can show that

$$-\lim_{t\to\infty}\frac{d\ln\mathcal{F}_{fl}(C^t)}{dt} \geq 4,$$

indicating an asymptotic decrease in $\mathcal{F}_{fl}(C^t)$ faster than e^{-4t}, independent of the target. We can also prove the finite time bound

$$\mathcal{F}_{fl}(C^t) \leq \mathcal{F}_{fl}(C^0) e^{-\left[\frac{2t}{\mathrm{tr}(\Lambda^{-1})(\mathrm{tr}(\Lambda)+|\mathrm{tr}((C^0)^{-1}-\Lambda)|)}\right]}.$$

The degenerate case $\mathbf{N < D + 1}$

Additionally, we can show the following result for the fixed points:

Theorem 3. *Given a D-dimensional multivariate Gaussian target density $p(x) = \mathcal{N}(x|\mu, \Sigma)$, using Algorithm 2 with $N < D + 1$ particles, the empirical mean converges to the exact mean μ. The $N - 1$ non-zero eigenvalues of C^t converge to a subset of the target covariance Σ spectrum. Furthermore, the **global minimum** of the regularised version $\widetilde{\mathcal{F}}$ of the free energy (17) corresponds to the **largest** eigenvalues of Σ.*

Proof. The proof is given in Appendix G. □

This result suggests that C^t might typically converge to an optimal low-rank approximation of Σ. We show an empirical confirmation in Section 4.2 for this conjecture. This suggests that it makes sense to apply our algorithm to high-dimensional problems even when the number of particles is not large. If the target density has significant support close to a low-dimensional submanifold, we might still obtain a reasonable approximation.

3.5. Structured Mean-Field

For high-dimensional problems, it may be useful to restrict the variational Gaussian approximation to the posterior to a specific structure via a structured mean-field approximation. In this way, spurious dependencies between variables that are caused by finite-sample effects could be explicitly removed from the algorithms. This is most easily incorporated in our approach by splitting a given collection of latent variables x into M disjoint subsets $x^{(i)}$. We reorder the vector indices in such a way that the first components correspond to $x^{(1)}, x^{(2)}$, and so on. Hence, we obtain $x = \{x^{(1)}, x^{(2)}, \ldots, x^{(M)}\}$. A structured mean-field approach is enforced by imposing a block matrix structure for the update matrix $A_{MF} = A_{(1)} \oplus \cdots \oplus A_{(M)}$, where $\oplus$ is the direct sum operator. It is easy to see that this construction corresponds to a related block structure of the Γ matrix in Equation (3). This means that the subsets of the random vectors are modeled as independent. Hence, when the number of particles grows to infinity, one recovers the fixed-point equations for the optimal MF structured Gaussian variational approximation from our approach. As previously, as the number of particles grows to infinity, we recover the optimal MF Gaussian variational approximation. Note that using a structured MF does not change the complexity of the algorithm but requires fewer particles to obtain a full-rank solution.

3.6. Comparison with SVGD

Given the similarities with the SVGD methods [24],one could question the differences of our approach. The model proposed by [10] using a *linear kernel* $k(x, x') = x^\top x' + 1$ has similar properties to our approach. The variable update becomes:

$$\begin{aligned}\frac{dx}{dt} &= \frac{1}{N}\sum_{i=1}^{N}(-k(x_i, x)\nabla\varphi(x_i) + \nabla_{x_i}K(x_l, x_i)) \\ &= \mathbb{E}_{\hat{q}}\left[I - \nabla\varphi(x)x^\top\right]x - \mathbb{E}_{\hat{q}}[\nabla\varphi(x)]\end{aligned}$$

The fixed points are

$$\begin{aligned}0 =& \mathbb{E}_{\hat{q}}[\nabla\varphi(x)] \\ I =& \mathbb{E}_{\hat{q}}\left[\nabla\varphi(x)x^\top\right] = \mathbb{E}_{\hat{q}}\left[\nabla\varphi(x)(x-m)^\top\right]\end{aligned}$$

where the last equality holds since $\mathbb{E}_{\hat{q}}[\nabla\varphi(x)] = 0$. This is the same as our algorithm fixed points (9). Similarly to Theorem 1, $D+1$ particles will converge to the exact D-dimensional multivariate Gaussian target. However, the generated flows are different. The main difference is that *we normalize our flow via the L_2 norm*, whereas [10] rely on the *reproducing kernel Hilbert space (RKHS) norm*, i.e., $\|\varphi\|_k^2 = \varphi^\top K^{-1}\varphi$ where $\varphi_i = \varphi(x_i)$ and $K_{ij} = k(x_i, x_j)$. For a full introduction on RKHS, we recommend [34]. Remarkably, centering the particles on the mean, namely, using the modified linear kernel $k(x, x') = (x - m)^\top (x' - m) + 1$, leads to the same dynamics. Additionally, when using SVGD, there is no direct possibility of computing the current KL divergence between the variational distribution and the target, unless some values are accumulated [35]. There is also no clear theory explaining what happens when the number of particles is smaller than the number of dimensions, for both distance-based kernels and the linear kernel.

4. Experiments

We now evaluate the efficiency of GPF and GF. First, given a Gaussian target, we compare the convergence of our approach with popular VGA methods, which are all described in Section 2. Second, we evaluate the effect of varying the number of particles for both Gaussian targets and non-Gaussian targets, especially with a low-rank covariance. Then, we evaluate the efficiency of our algorithm on a range of real-world binary classification problems through a Bayesian logistic regression model and a series of BNN on the MNIST dataset.

All the Julia [36] code and data used to reproduce the experiments are available at the Github repository: https://github.com/theogf/ParticleFlow_Exp (accessed on 27 July 2021).

4.1. Multivariate Gaussian Targets

We consider a 20-dimensional multivariate Gaussian target distribution. The mean is sampled from a normal Gaussian $\mu \sim \mathcal{N}(0, I_D)$ and the covariance is a dense matrix defined as $\Sigma = U\Lambda U^\top$, where U is a unitary matrix and Λ is a diagonal matrix. Λ is constructed as $\log_{10}(\Lambda_{ii}) = \frac{\log_{10}(\kappa)(i-1)}{D-1} - 1$ where κ is the condition number, i.e., $\kappa = \Lambda_{\max}/\Lambda_{\min}$. This means that, for $\kappa = 1$, we obtain a $\Sigma = 0.1\mathbb{I}$, and for $\kappa = 100$, we obtain eigenvalues ranging uniformly from 0.1 to 10 in log-space.

We compare GPF and GF to the state-of-the art methods for VGA described in Section 2, namely *Doubly Stochastic VI* **(DSVI)** [14], *Factor Covariance Structure* **(FCS)** [15] with rank $p = D$, *iBayes Learning Rule* **(IBLR)** [17] with a full-rank covariance and their Hessian approach, and Stein Variational Gradient Descent with both a linear kernel (**Linear SVGD**) [10] and a squared-exponential kernel (**Sq. Exp. SVGD**) [24]. For all methods, we set the number of particles or, alternatively, the number of samples used by the estimator, as $D+1$, and use standard gradient descent ($x^{t+1} = x^t + \eta\varphi^t(x^t)$) with a learning rate of $\eta = 0.01$ for all particle methods. We use RMSProp [37] with a learning rate of 0.01 for all stochastic methods. We run each experiment 10 times with 30,000 iterations, and plot the average error on the mean and the covariance with one standard deviation. For GPF, we additionally evaluate the method with and without using natural gradients for the mean (i.e., pre-multiplying the averaged gradient with C^t), indicated, respectively, with a dashed and solid line. Figure 2 reports the L_2 norm of the difference between the mean and covariance with the true posterior over time for the target condition number $\kappa \in \{1, 10, 100\}$.

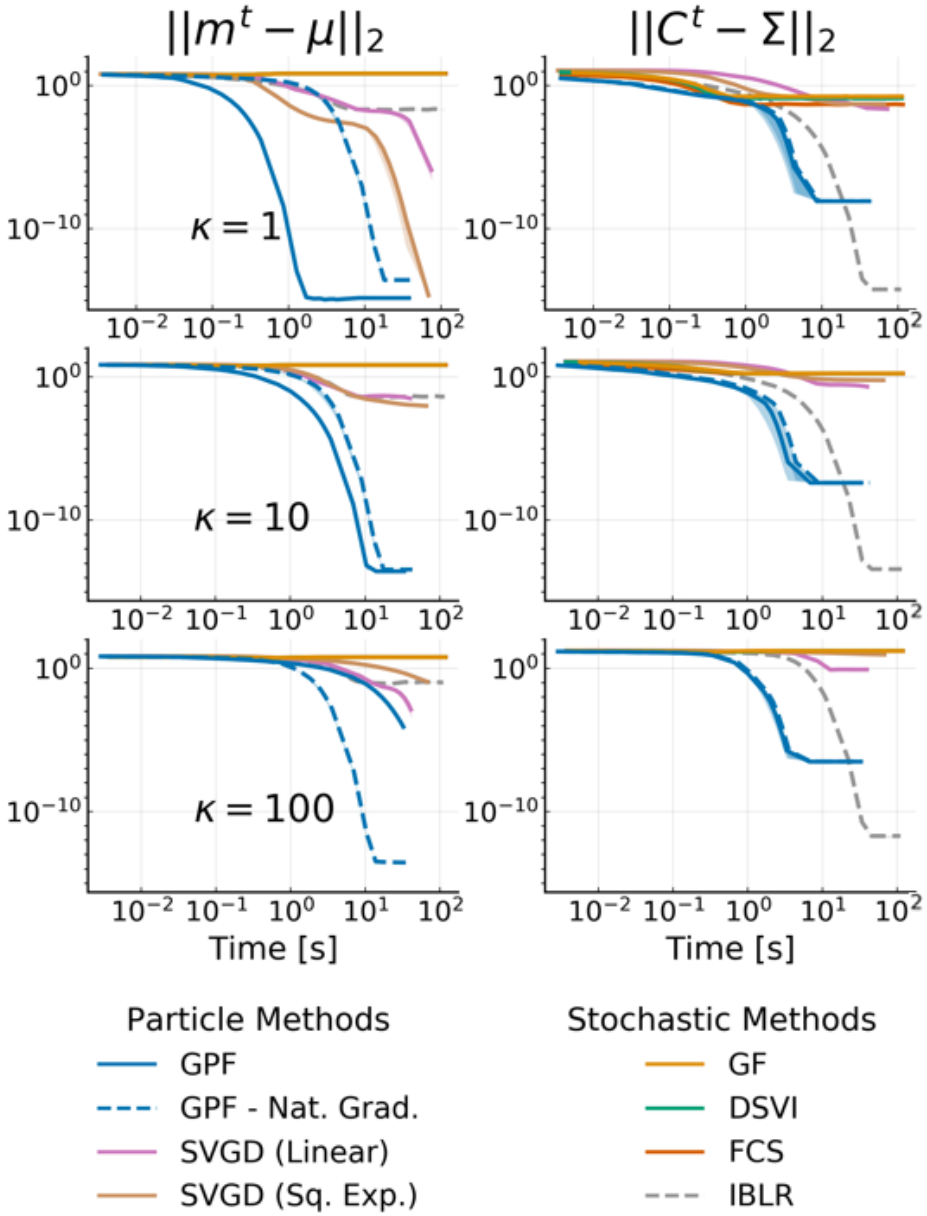


Figure 2. L^2 norm of the difference between the target mean μ (left side) and target covariance Σ (right side) with the inferred variational parameters m^t and C^t against time for 20-dimensional Gaussian targets with condition number κ. We use $D + 1$ particles/samples and show the mean over 10 runs as well as the 68% credible interval. Methods with dashed curves use natural gradients on the mean. Note that DSVI, GF and FCS are overlapping and are, at this scale, indistinguishable from one another.

As Theorem 1 predicts, GPF converges exactly to the true distribution, regardless of the target. GF and other methods based on stochastic estimators cannot obtain the same precision as their accuracy is penalized by the gradient noise. IBLR approximate the covariance perfectly, despite the stochasticity of its estimator; however IBLR needs to compute the true Hessian at each step. When using a Hessian approximation instead, IBLR performed just like DSVI; the true benefit of IBLR appears when second-order functions are computed, which is naturally intractable in high-dimensions. SVGD with a linear kernel, achieves a good performance but is highly unstable: most of the runs (ignored here) diverge. This is due to the dot computation $x^\top x$ which can become extremely high, especially for non-centered data. For this reason, we do not consider this method for the later experiments. SVGD with a sq. exp. kernel obtains a good estimate for the mean but fails to approximate the covariance.

Perhaps surprisingly, GF does not perform much better than DSVI or FCS. This is potentially due to *the benefit of Riemannian gradients being canceled by the gradient noise* [38] providing a strong argument for particle-based methods over stochastic estimators.

Remarkably, we also confirm Theorem 2, that the convergence speed of C^t is independent of the target Σ, while the convergence speed of m^t has this dependency unless the natural gradient is used (see the dashed curves). The case $\kappa = 1$ highlights that *natural gradient do not necessarily improve convergence speed*.

4.2. Low-Rank Approximation for Full Gaussian Targets

We explore the effect of the number of particles for both Gaussian and non-Gaussian targets. We use the same Gaussian target from the previous experiment in 50 dimensions with a full-rank covariance determined by their condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$. The covariance eigenvalues λ_i in log-space range uniformly from 0.1 to 0.1κ. For a given target multivariate Gaussian, we vary the number of particles from 2 to $D + 1$ and look at the absolute difference of $|\mathrm{tr}(C - \Sigma)|$. The results in $D = 50$, as well as the corresponding predictions (in dashed-black), from Theorem 3, are shown on Figure 3.

The empirical results perfectly match the theoretical predictions, confirming that, for Gaussian targets, the particles determine a low-rank approximation whose spectrum is equal to the largest eigenvalues from the target.

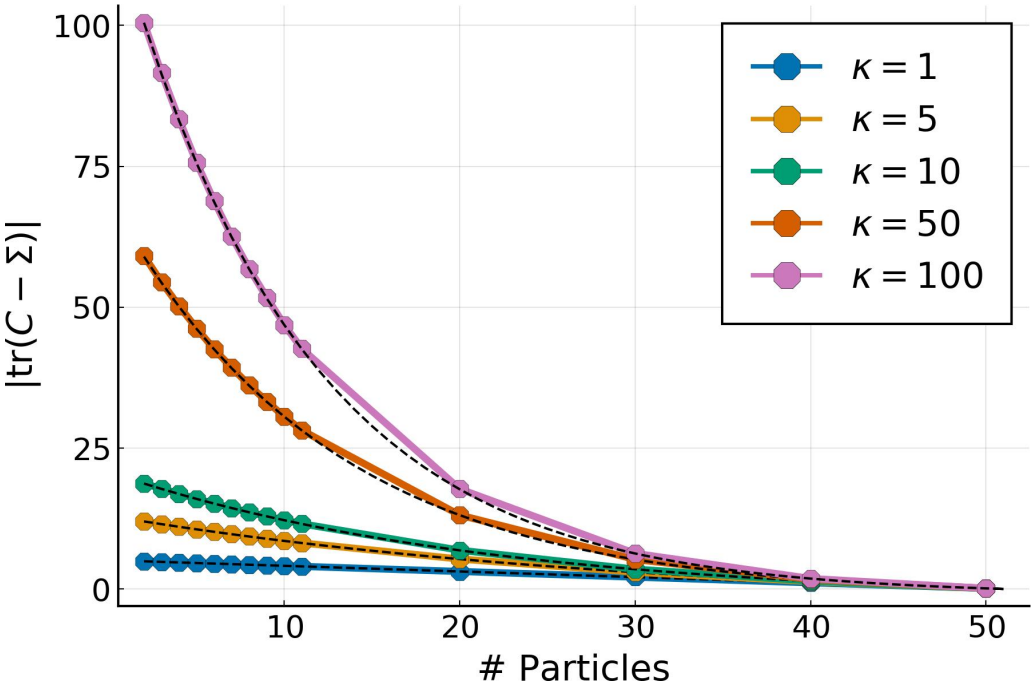


Figure 3. Trace error for a Gaussian target with $D = 50$ and condition numbers κ for a varying number of particles with GPF. Predictions from Theorem 3 are shown in dashed-black.

4.3. High-Dimensional Low-Rank Gaussian Targets

We consider a typical low-rank target case where the dimensionality is high but the effective rank of the covariance is unknown. The target is given by $p(x) = \mathcal{N}(\mu, \Sigma)$ where $\mu \sim \mathcal{N}(0, \mathbb{I}_D)$, the covariance is defined by $\Sigma = U\Lambda U^\top$, where U is a $D \times D$ unitary matrix and Λ is a diagonal matrix defined by

$$\Lambda_{ii} = \begin{cases} \mathcal{N}(2,1), & \text{if } i \leq K \\ 10^{-8}, & \text{otherwise} \end{cases}$$

where K is the effective rank of the target. We pick $D = 500$ and vary $K \in \{10, 20, 30\}$ to simulate a true problem where the correct K is not known. We test all methods allowing

for low-rank structure, namely, GPF, GF, FCS and SVGD (Linear and Sq. Exp.). We fix the rank (or the number of particles) to be 20; therefore, we obtain three cases where the rank is exact, under-estimated, and over-estimated. For all methods, we use RMSProp [37] for the stochastic methods, or a diagonal version of it (see Section 4.4) for the particle ones. The error of the mean and the covariance is shown in Figure 4. Note that the difference in the initial error on the covariance is due to the difficulty of starting with the same covariance between particle and stochastic methods.

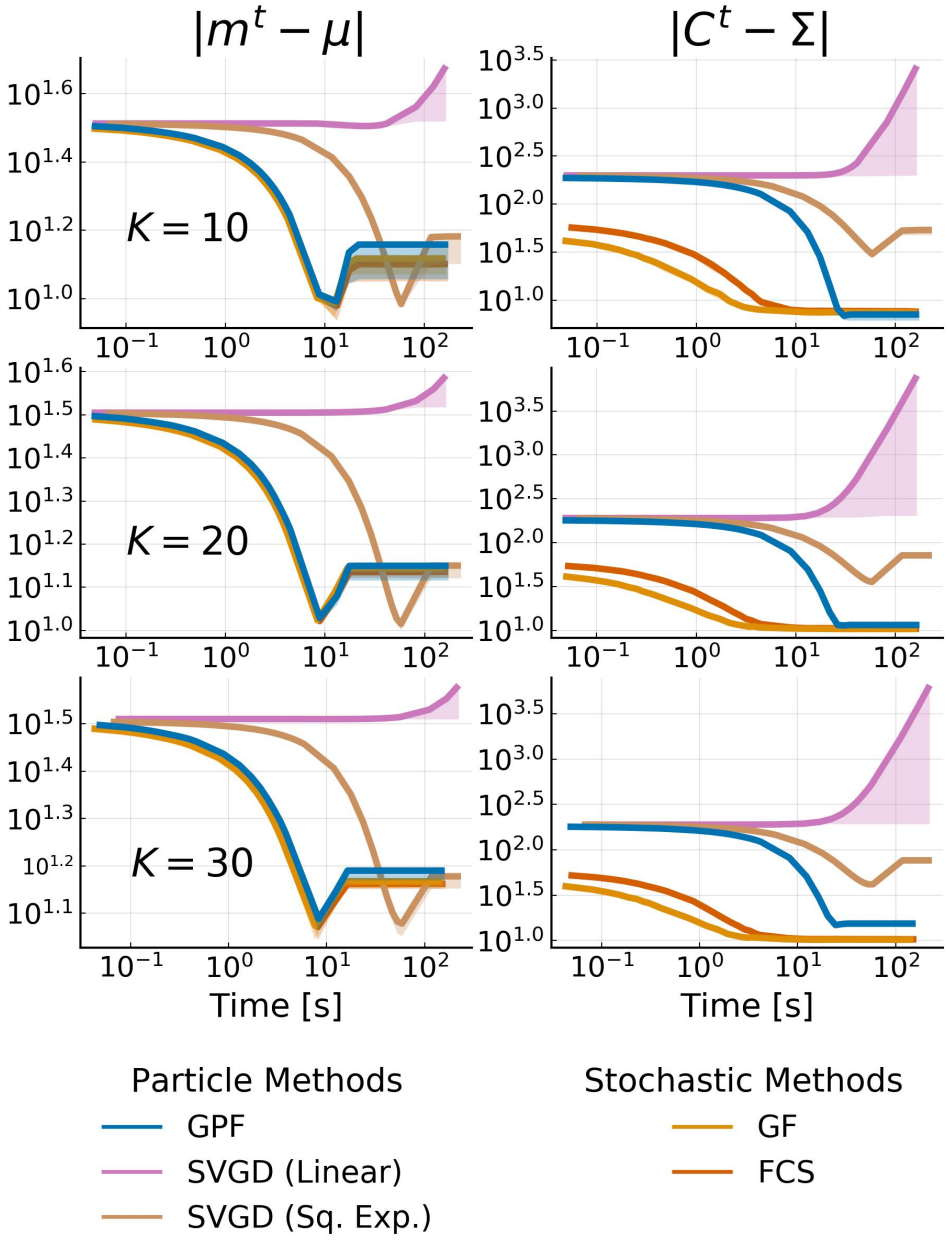


Figure 4. Convergence plot of low-rank methods for a 500-dimensional multivariate Gaussian target with effective rank $K \in \{10, 20, 30\}$. The rank of each method is fixed as 20. The difference in the starting point for the covariance is due to the initialization difference between each method. We show the mean over 10 runs for each method with shadowed areas representing the 68% credible interval.

We observe once again that the SVGD with a linear kernel fails to converge due to the large gradients. All methods perform equally in the estimation of the mean while being non-influenced by the rank of the target. As expected, the approximation quality for the covariance degrades when the rank gets bigger, but all algorithms still converge to good

approximations. SVGD with a sq. exp. kernel performs much worse than the rest of the methods. This is a known phenomenon where, for high dimensions, the covariance SVGD is either over- or underestimated.

4.4. Non-Gaussian Target

We now investigate the behavior of our algorithm with non-Gaussian target distributions. We built a two-dimensional banana distribution: $p(x) \propto \exp(-0.5(0.01x_1^2 + 0.1(x_2 + 0.1x_1^2 - 10)^2))$, varied the number of particles used for GPF in $\{3, 5, 10, 20, 50\}$ and compared it with a standard full-rank VGA approach. We also showed the impact of replacing a fixed η with the Adam [39] optimizer for 50 particles. The results are shown in Figure 5. As expected, increasing the number of particles madesthe distribution obtained via GPF increasingly closer to the optimal standard VGA, even in a non-Gaussian setting. However, using a momentum-based optimizer such as Adam breaks the linearity assumption of the original flow (5) and leads to a twisted representation of the particles. (We observed the same behavior with other momentum-based optimizers). A simple modification of the most known optimizers allows the linearity to be maintained while correctly adapting the learning rate to the shape of the problem. Most optimisers accumulate momentum or gradients element-wise, and end up modifying the updates as $x^{t+1} = x^t + P^t \odot \varphi^t(x^t)$, where $P^t \in \mathbb{R}^{D\times D}$ is the preconditioner obtained via the optimiser and $\odot$ is the Hadamard product. By instead taking the average over each dimensions, we obtained the updates $x^{t+1} = x^t + P^t\varphi^t(x^t)$, where P^t is a $D \times D$ diagonal matrix. The details of the dimension-wise conditioners for ADAM, AdaGrad and AdaDelta are given in Appendix H.

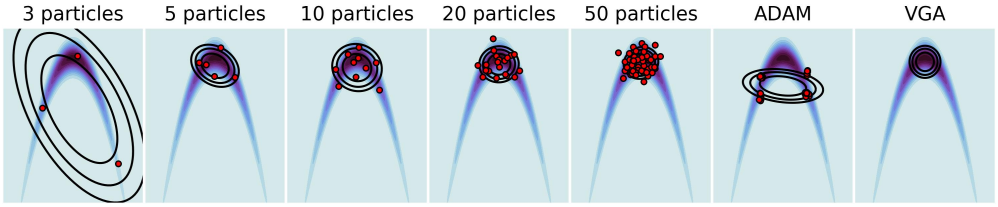
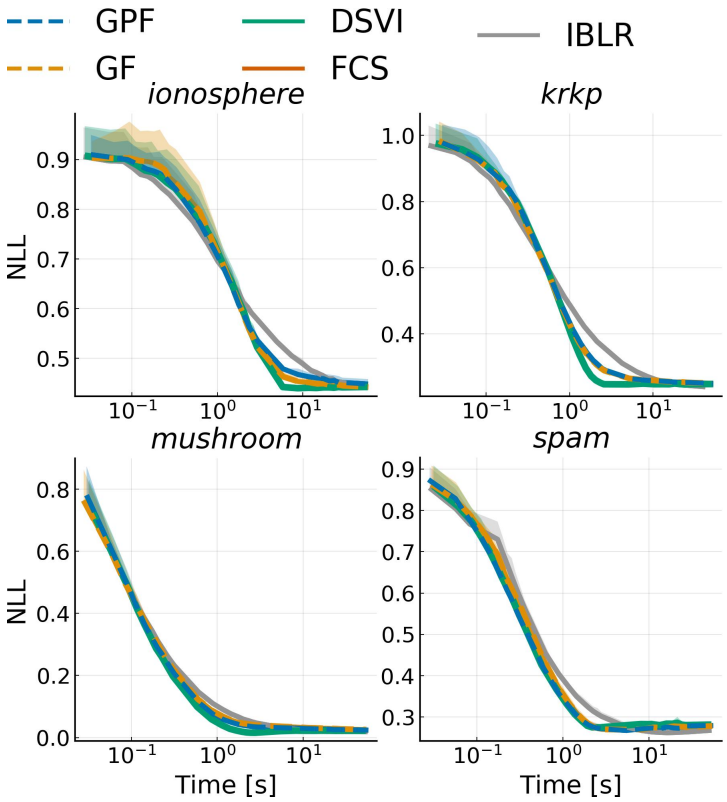


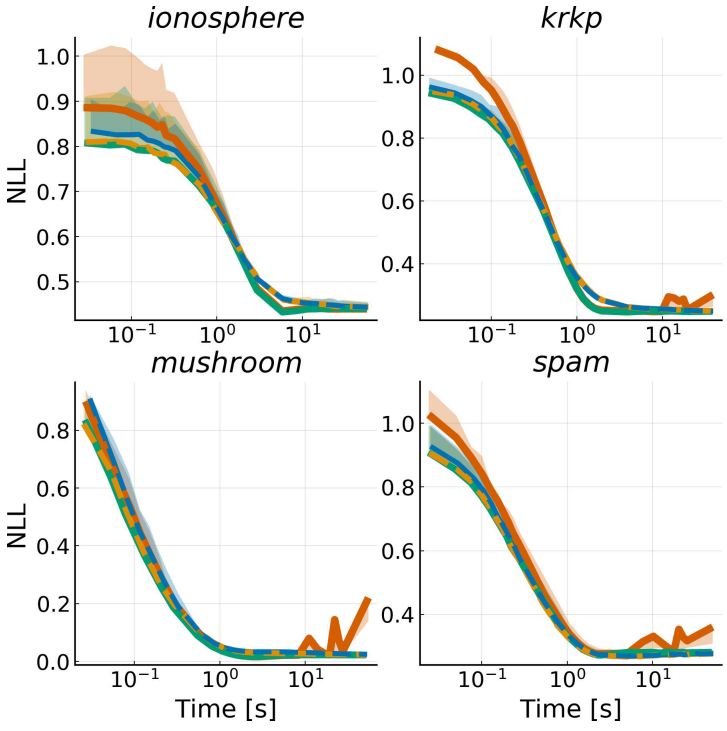
Figure 5. Two-dimensional Banana distribution. Comparison of GPF using an increasing number of particles and a different optimizer (ADAM) with the standard VGA (rightmost plot).

4.5. Bayesian Logistic Regression

Finally, we considered a range of real-world binary classification problems modeled with a Bayesian logistic regression. Given some data $\{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathbb{R}^D$ and $y \in \{-1, 1\}$, we defined the model $y_i \sim \text{Bernoulli}(\sigma(w^\top x_i))$ with weight $w \in \mathbb{R}^D$, and with σ being the logistic function. We set a prior on w: $w\ \mathcal{N}(0, 10\mathbb{I}_D)$. We benchmarked the competing approaches over four datasets from the UCI repository [40]: `spam` ($N = 4601, D = 104$), `krkp` ($N = 351, D = 111$), `ionosphere` ($N = 3196, D = 37$) and `mushroom` ($N = 8124, D = 95$). We ran all algorithms discussed in Section 4.1, both with and without a mean-field approximation; SVGD was omitted since it is too unstable. All algorithms were run with a fixed learning rate $\eta = 10^{-4}$, and we used mini-batches of size 100. We show alternative training settings in Appendix I. Note that FCS, for mean-field, simplifies to DSVI Additionally, we did not consider full-rank IBLR, as it is too expensive, and we used their reparametrized gradient version for the Hessian. Figure 6 shows the average negative log-likelihood on 10-fold cross-validation with one standard deviation for each dataset. While, as expected, the advantages shown for Gaussian targets do not transfer to non-Gaussian targets, GPF and GF are consistently on par with competitors. On the other hand, IBLR tends to be outperformed. It is also interesting to note that mean-field does not seem to have a negative impact on these problems, and performance remains the same even with a full-rank matrix.



(**a**) Mean-field approximation



(**b**) No mean-field approximation

Figure 6. Average negative log-likelihood vs. time on a test-set over 10 runs against training time for a Bayesian logistic regression model applied to different datasets. Top plots use a mean-field approximation, while bottom plots use a low-rank structure for the covariance with rank $L = 100$.

4.6. Bayesian Neural Network

We ran our algorithm on a standard network with two hidden layers each, with $L = 200$ neurons and tanh activation functions (we additionally tried ReLU [41], but some baselines failed to converge). We trained on the MNIST dataset [42] ($N = 60{,}000$, $D = 784$) and used an isotropic prior on the weights $p(w) = \mathcal{N}(0, \alpha I_D)$ with $\alpha = 1.0$. We additionally compared these with *Stochastic Weight Averaging-Gaussian* **(SWAG)** [27] with an SGD learning rate of 10^{-6} (selected empirically) and *Efficient Low-Rank Gaussian Variational Inference* **(ELRGVI)** [26]. We varied the assumptions on the covariance matrix to be diagonal (**Mean-Field**), or to have rank $L \in \{5, 10\}$. Additionally, we showed, for GPF, the effect of using a structured mean-field assumption by imposing the independence of the weights between each layer (**GPF (Layers)**).

We trained each algorithm for 5000 iterations with a batchsize of 128($\sim$10 epochs) and reported the final average negative log-likelihood, accuracy and expected calibration error [43] on the test set ($N = 10{,}000$) on Table 1. The predictive distribution is given by

$$p(y = k|x^*, \mathcal{D}) = \int p(y = k|x^*, w)p(w|\mathcal{D})dw \approx \int p(y = k|x^*, w)q(w)dw,$$

where $\mathcal{D}$ is the training data, and x^* is a test sample. We computed the accuracy and the average negative test log-likelihood as:

$$\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} 1_{y_i}(\arg_k \max p(y = k|x_i^*, \mathcal{D}))$$

$$\text{NLL} = -\frac{1}{N}\sum_{i=1}^{N} \log p(y = y_i|x_i^*, \mathcal{D})$$

where $1_y(x)$ is the indicator function (equal to 1 for $y = x$, 0 otherwise). For the definition of expected calibrated error, we refer the reader to [43]. Additional convergence and uncertainty calibration plots can be found in Appendix I.

Table 1. Negative Log-Likelihood (NLL), Accuracy (Acc), and Expected Calibration Error (ECE) for a *Bayesian Neural Networks* **(BNN)** on the MNIST dataset. We varied the rank of the variational covariance from mean-field (all variables are independent) to a low-rank structure with $L \in \{5, 10\}$. Bold numbers indicated the best performance, and italic bold numbers indicate the best performance when restricted to VGA methods. Convergence and calibration plots can be found in Appendix I.

Alg.	**Mean-Field**			$L = 5$			$L = 10$		
	NLL	**Acc**	**ECE**	**NLL**	**Acc**	**ECE**	**NLL**	**Acc**	**ECE**
GPF	0.183	0.95	0.0384	0.166	***0.96***	0.0918	0.172	0.955	0.0869
GPF (Layers)	-	-	-	***0.147***	0.958	**0.0181**	0.178	0.952	0.0395
GF	0.178	0.953	0.0706	0.185	0.956	0.136	0.171	0.952	0.0455
DSVI	0.204	0.945	0.11	-	-	-	-	-	-
SVGD (Sq. Exp)	-	-	-	0.139	0.965	0.0732	**0.133**	**0.967**	0.0879
SWAG	-	-	-	0.257	0.957	0.0662	0.287	0.956	0.0878
ELRGVI	-	-	-	0.453	0.901	0.53	0.537	0.882	0.777

Overall, the SVGD method performed best in terms of both accuracy and negative log-likelihood. However, SVGD is not in the same category as others, since it is not a VGA. For VGAs, we observed that a low-rank approximation improves upon mean-field methods. In particular, assuming independence between layers provides a large advantage to GPF. GPF and GF generally perform equally or better than all the other VGA methods. Note that, although not reported here, all methods needed approximately the same time for the 5000 iterations, except for SWAG, which only needed the MAP and a few thousand iterations of SGD afterward, making it generally faster but also less controlled (a grid search was needed to find the appropriate learning for SGD).

5. Discussion

We introduced GPF, a general-purpose and theoretically grounded, particle-based approach, to perform inference with variational Gaussians as well as GF its parameter version. We were able to show the convergence of the particle algorithm based on an empirical approximation of the free energy. We also showed that we can approximate high-dimensional targets by allowing for low-rank approximations with a small number of particles. The results for Gaussian targets suggest that the convergence of posterior covariance approximation may relax asymptotically fast, with small dependence on the target. This work is the first step in analyzing convergence speed and guarantees in inference with variational Gaussians, and future work could extend guarantees to non-Gaussian problems. One could also take advantage of existing particle-based VI methods to accelerate inference further or reach a better optima [44,45].

Author Contributions: Conceptualization, T.G.-F. and M.O.; methodology, T.G.-F., V.P. and M.O.; software, T.G.-F.; validation, T.G.-F.; formal analysis, T.G.-F.; investigation, T.G.-F.; resources, T.G.-F. and V.P.; data curation, T.G.-F.; writing—original draft preparation, T.G.-F., V.P. and M.O.; writing—review and editing, T.G.-F., V.P. and M.O.; visualization, T.G.-F.; supervision, M.O.; project administration, T.G.-F.; funding acquisition, M.O. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Data Availability Statement: Datasets can be found on the UCI dataset website [40] and the MNIST dataset can be found on Yann Lecun website [42].

Acknowledgments: We thank Fela Winkelmolen for his initial help on computations, Jannik Thümmel for his work on the linear SVGD and the reviewers for their insightful comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of the Optimal Parameters

In Section 3, we considered the optimization problem:

$$\min_{A^t, b^t \in \mathcal{B}} \frac{d\mathcal{F}[q^t]}{dt} \text{ where } \mathcal{B} = \{A^t, b^t : \|A^t\|_F^2 = 1,\ \|b^t\|^2 = 1\},$$

where we have introduced $\|A^2\|_F^2 = \mathrm{tr}(AA^\top)$, the Froebius norm and $\|b^t\|$, the L_2 norm and

$$\frac{d\mathcal{F}[q^t]}{dt} = -\mathrm{tr}\left[A^t (A_\star^t)^\top\right] - (b^t)^\top b_\star^t \tag{A1}$$

To solve this problem, we used the Lagrange multiplier method. We write the Lagrangian as:

$$\mathcal{L}(A^t, b^t) = \frac{d\mathcal{F}[q^t]}{dt} - \lambda_A g(A^t) - \lambda_b h(b^t),$$

where $g(A) = \mathrm{tr}(AA^\top) - 1$ and $h(b) = \|b\|_2^2 - 1$. For simplicity we can divide the problem as:

$$\begin{aligned}\mathcal{L}(A^t) &= -\mathrm{tr}\left[A^t (A_\star^t)^\top\right] - \lambda_A g(A^t)\\ \mathcal{L}(b^t) &= -(b^t)^\top b_\star^t - \lambda_b h(b^t)\end{aligned}$$

For A^t, we have the constraints:

$$\begin{aligned}\nabla_{A^t}\mathrm{tr}\left[A^t(A^t_\star)^\top\right] &= \lambda_A \nabla_{A^t} g(A^t)\\ g(A^t) &= 0\end{aligned}$$

Computing the gradients is straightforward:

$$\begin{aligned}A^t_\star &= 2\lambda_A A^t\\ \Rightarrow A^t &= \frac{A^t_\star}{2\lambda_A}\\ \Rightarrow \frac{1}{4\lambda_A^2}\mathrm{tr}(A^t_\star(A^t_\star)^\top) &= 1\\ \Rightarrow \lambda_A &= \sqrt{\frac{\mathrm{tr}(A^t_\star(A^t_\star)^\top)}{4}}.\end{aligned}$$

which gives us the result $A^t = \frac{A^t_\star}{\|A^t_\star\|_F}$. Similarly for b^t:

$$\begin{aligned}\nabla_{b^t}(b^t)^\top b^t_\star &= \lambda_b \nabla_{b^t} h(b^t)\\ h(b^t) &= 0.\end{aligned}$$

Replacing the gradients gives:

$$\begin{aligned}b^t_\star &= 2\lambda_b b^t\\ \Rightarrow b^t &= \frac{b^t_\star}{2\lambda_b}\\ \Rightarrow \frac{1}{4\lambda_b^2}\|b^t_\star\|_2^2 &= 1\\ \Rightarrow \lambda_b &= \frac{2}{\|b^t_\star\|_2}\end{aligned}$$

which gives us the result $b^t = \frac{b^t_\star}{\|b^t_\star\|_2}$.

Appendix B. Relaxation of the Empirical Free Energy

We prove the decrease in the empirical free energy (17) under the particle flow when the covariance C is nonsingular. We define the empirical distribution $\hat{q}(x) = \frac{1}{N}\sum_{i=1}^N \delta_{x,x_i}$ with a finite number N of particles. The empirical free energy is defined as

$$\mathcal{F}[\hat{q}] = \mathbb{E}_{\hat{q}}[\varphi(x)] - \frac{1}{2}\log|C|.$$

We are interested in the temporal change of the free energy, when particles move under a general linear dynamics

$$\frac{dx_i}{dt} = b + A(x_i - m).$$

The induced dynamics for $\mathcal{F}$ are:

$$\frac{d\mathcal{F}}{dt} = \mathbb{E}_{q^t}\left[\nabla_x\varphi(x)^\top\frac{dx}{dt}\right] - \frac{1}{2}\mathrm{tr}(C^{-1}\frac{dC}{dt})$$

For notational simplicity, we introduce $g(x) = \nabla_x\varphi(x)$ and $\dot{x} = \frac{dx}{dt}$ (similarly $\dot{m} = \frac{dm}{dt}$).

$$\begin{aligned}\frac{dC}{dt} =& \frac{d}{dt}\mathbb{E}_q\left[(x-m)(x-m)^\top\right]\\ =& \mathbb{E}_q\left[(\dot{x}-\dot{m})(x-m)^\top\right] + \mathbb{E}_q\left[(x-m)(\dot{x}-\dot{m})^\top\right]\\ =& \mathbb{E}_q\left[\dot{x}x^\top + x\dot{x}^\top - \dot{m}m^\top - m\dot{m}^\top\right]\\ =& \mathbb{E}_q\left[\dot{x}(x-m)^\top\right] + \mathbb{E}_q\left[(x-m)\dot{x}^\top\right]\end{aligned}$$

$$\begin{aligned}\frac{d\mathcal{F}}{dt} =& \mathbb{E}_q\left[g(x)^\top\dot{x}\right] -\\ & \frac{1}{2}\mathbb{E}_q\left[\mathrm{tr}(C^{-1}\dot{x}(x-m)^\top) + \mathrm{tr}(C^{-1}(x-m)^\top\dot{x}^\top)\right]\\ =& \mathbb{E}_q\left[\dot{x}^\top\left(g(x) - C^{-1}(x-m)\right)\right]\end{aligned} \tag{A2}$$

where we used the permutation properties of the trace.

Plugging the dynamics into Equation (A2), we obtain:

$$\begin{aligned}\frac{d\mathcal{F}}{dt} =& b^\top\mathbb{E}_q[g(x)] + \mathbb{E}_q\left[(x-m)^\top A^\top g(x)\right]\\ & - \mathbb{E}_q\left[(x-m)^\top A^\top C^{-1}(x-m)\right]\end{aligned} \tag{A3}$$

where we used the fact that $b^\top C^{-1}\mathbb{E}_q[x-m] = 0$.

We next look for conditions on b and A, under which $\frac{d\mathcal{F}}{dt} < 0$, i.e., the dynamics will lead to a decrease in the free energy. We pick $b = -\beta_1\mathbb{E}_q[g(x)]$, where $\beta_1 > 0$, and we obtain, for the first term in (A3):

$$-\beta_1\|\mathbb{E}_q[g(x)]\|^2 \leq 0.$$

For A, let us first define $\psi = \mathbb{E}_q\left[g(x)(x-m)^\top\right]$ and rewrite the second and last term of the Equation (A3) as:

$$\begin{aligned}\mathbb{E}_q\left[(x-m)^\top A^\top g(x)\right] =& \mathrm{tr}\left(\mathbb{E}_q\left[A^\top g(x)(x-m)^\top\right]\right)\\ =& \mathrm{tr}\left(A^\top\psi\right)\\ \mathbb{E}_q\left[(x-m)^\top A^\top C^{-1}(x-m)\right] =& \mathrm{tr}\left(A^\top C^{-1}C\right)\\ =& \mathrm{tr}(A)\end{aligned}$$

Combining both, we get $\mathrm{tr}\left(A^\top(\psi - I)\right)$. Similarly to the previous step, we pick $A = -\beta_2(\psi - I)$, where $\beta_2 \geq 0$, which leads to another negative term:

$$-\beta_2\mathrm{tr}((\psi - I)^\top(\psi - I)) \leq 0,$$

where we use the fact that $X^\top X$ is a positive semi-definite matrix for any real valued X.

Note that different forms of A (e.g., β_2 are replaced by a positive definite matrix) could be used, as long as the trace of the product stays positive. Inserting b and A, the free energy dynamics become

$$\frac{d\mathcal{F}}{dt} = -\beta_1\|\mathbb{E}_q[g(x)]\|^2 - \beta_2\mathrm{tr}((\psi - I)^\top(\psi - I))$$

The variable dynamics are given by

$$\begin{aligned}\frac{dx}{dt} &= -\beta_1 \mathbb{E}_q[g(x)] - \beta_2(\psi - I)(x - m) \\ &= -\beta_1 \mathbb{E}_q[g(x)] \\ &\quad - \beta_2\Big(\mathbb{E}_q\Big[g(x)(x-m)^\top\Big] - I\Big)(x-m),\end{aligned}$$

which is equivalent to Equation (5), for $\beta_1 = \beta_2 = 1$. Our result shows that the empirical approximation of the free energy decreases under the particle flow.

Appendix C. Riemannian Gradient for Matrix Parameter Γ

The parameter flow for the matrix Γ in (11) is given by

$$\frac{d\Gamma^t}{dt} = \Gamma^t - \mathbb{E}_{q^0}\Big[\nabla_x \varphi(x^t)(x^0 - m^0)^\top\Big]\Gamma^t(\Gamma^t)^\top.$$

This is easily rewritten in terms of the parameter gradient as $\frac{d\Gamma^t}{dt} = \frac{\partial \mathcal{F}}{\partial \Gamma}\Gamma\Gamma^\top$

Similar to natural gradients, which are defined by the metric, which is induced by the Fisher–matrix, we can rewrite the parameter change in terms of a different *Riemannian* gradient. This gradient is the direction of change $d\Gamma = \Gamma(t + dt) - \Gamma(t)$, which yields the steepest descent of the free energy over a small time interval dt. As an extra condition, one keeps the length of $d\Gamma$ (measured by a 'natural' metric, which has specific invariance properties) fixed. This is defined by an inner product (the squared length) $\langle d\Gamma, d\Gamma\rangle_\Gamma$ in the tangent space of small deviations $d\Gamma$ from the matrix Γ. Hence, $d\Gamma$ is found by minimising $\mathcal{F}(\Gamma(t) + d\Gamma, m)$ (for small $d\Gamma$) under the condition that $\langle d\Gamma, d\Gamma\rangle_{\Gamma(t)}$ is fixed. Following [20] (Theorem 6), a natural metric in the space of symmetric nonsingular matrices can be defined as

$$\langle d\Gamma, d\Gamma\rangle_\Gamma \doteq \mathrm{tr}\Big((d\Gamma\ \Gamma^{-1})^\top d\Gamma\ \Gamma^{-1}\Big).$$

This metric is invariant against multiplications of Γ and $d\Gamma$ by matrices Y, i.e., $\langle d\Gamma, d\Gamma\rangle_\Gamma = \langle d\Gamma\ Y, d\Gamma\ Y\rangle_{\Gamma Y}$ and reduces to the Euclidian metric at the unit matrix $\Gamma = I$.

The direction of the natural gradient is obtained by expanding the free energy for small $d\Gamma$ and introducing a Lagrange–multiplier λ for the constraint. One ends up with the quadratic form

$$\frac{\partial \mathcal{F}}{\partial \Gamma} d\Gamma + \lambda \mathrm{tr}\Big((d\Gamma\ \Gamma^{-1})^\top d\Gamma\ \Gamma^{-1}\Big)$$

to be minimised by $d\Gamma$. By taking the derivative with respect to $d\Gamma$, one finds that the direction of $d\Gamma$ agrees with the right equation of the flow (11).

Appendix D. Regularised Free Energy for $N \leq D$

The problem of defining an empirical approximation for $N \leq D$ particles is that the empirical covariance becomes singular and typically has $N - 1$ nonzero eigenvalues, and thus $|C| = 0$. Note that the extra 0 eigenvalue is derived from the fact that the empirical sum of fluctuations must be zero, which provides an additional linear constraint.

We can regularise the log determinant term by replacing the zero eigenvalues of C: $\lambda_i = 0 \to \tilde{\lambda}_i = 1$. The new covariance $\tilde{C}$ becomes

$$\log|\widetilde{C}| = \sum_{i:\lambda_i > 0} \log \lambda_i,$$

since $\log 1 = 0$. The dynamics of the particles stays the same. To rewrite this formally in terms of matrices, we define

$$\widetilde{C} = C + C_\perp$$

where

$$C_\perp = \sum_{i:\lambda_i=0} e_i e_i^\top$$

and $e_i = i$th eigenvector of C. This replaces all 0 eigenvalues by 1. $C_\perp$ is a projector: $C_\perp^2 = C_\perp$ and $C_\perp(I - C_\perp) = 0$. We also have $\mathrm{tr}(C_\perp) = D - (N-1)$. In the following, it is useful to introduce the $D \times N$ matrix of fluctuations Z, such that $C = ZZ^\top/N$. The column vectors of Z span the subspace of eigenvectors e_i with $\lambda_i > 0$. Hence, it follows that $C_\perp Z = 0$.

We want to show that the regularised free energy $\widetilde{F}$ decreases under the particle dynamics for $N \leq D$. Since the part of the time derivative of $\widetilde{F}$ that depends on $\frac{dm}{dt}$ is not changed, we will only discuss the fluctuation part in the following.

It is useful to introduce the matrix:

$$\widetilde{A} \doteq I - C_\perp - gZ^\top/N = A - C_\perp,$$

with $g = \nabla_x \varphi(x)$ is the $D \times N$ matrix of the gradient.

$$\begin{aligned}\mathbb{E}_q\left[g(x)^\top \frac{dx}{dt}\right] =& \mathrm{tr}(A) - \mathrm{tr}(A^\top A)\\ =& \mathrm{tr}(\widetilde{A} + C_\perp) - \mathrm{tr}((\widetilde{A} + C_\perp)^\top(\widetilde{A} + C_\perp))\\ =& \mathrm{tr}(\widetilde{A}) - \mathrm{tr}(\widetilde{A}^\top \widetilde{A}).\end{aligned}$$

To obtain this result, we need

$$\begin{aligned}\mathrm{tr}(C_\perp \widetilde{A}) =& \mathrm{tr}(C_\perp \widetilde{A}^\top)\\ =& \mathrm{tr}(C_\perp(I - C_\perp) - C_\perp Z g^\top/N) = 0.\end{aligned}$$

We need to work out

$$\begin{aligned}-\frac{1}{2}\frac{d\ln|\widetilde{C}|}{dt} =& -\frac{1}{2}\mathrm{tr}\left(\frac{d\widetilde{C}}{dt}\widetilde{C}^{-1}\right)\\ =& -\frac{1}{2}\mathrm{tr}\left(\frac{dC}{dt}\widetilde{C}^{-1}\right)\end{aligned}$$

where we have used the fact that the eigenvalues $\lambda_i = 1$ of $\widetilde{C}$ have a zero time derivative and can be omitted. We use the linear dynamics $\frac{dZ}{dt} = AZ$ to obtain:

$$\begin{aligned}\frac{dC}{dt} =& = CA^\top + AC\\ =& (\widetilde{C} - C_\perp)(\widetilde{A}^\top + C_\perp) + (\widetilde{A} + C_\perp)(\widetilde{C} - C_\perp)\\ =& \widetilde{C}\widetilde{A}^\top + \widetilde{A}\widetilde{C} + C_\perp\widetilde{C} + \widetilde{C}C_\perp - \widetilde{A}C_\perp - C_\perp\widetilde{A}^\top - 2C_\perp\\ =& \widetilde{C}\widetilde{A}^\top + \widetilde{A}\widetilde{C},\end{aligned}$$

where we have used $C_\perp^2 = C_\perp$ and $C_\perp \widetilde{A}^\top = 0$. Hence

$$-\frac{1}{2}\mathrm{tr}\left(\frac{d\widetilde{C}}{dt}\widetilde{C}^{-1}\right) = -\mathrm{tr}(\widetilde{A}).$$

Finally, the temporal change in the free energy due to the fluctuations is given by

$$\frac{d\widetilde{\mathcal{F}}}{dt} = -\mathrm{tr}(\widetilde{A}^\top \widetilde{A}) \leq 0.$$

Note that this proof is not only valid for $N \leq D$, but also for $N > D$, as the overall computations are simplified with $C_\perp = 0$. A more detailed proof for $N > D$ is, furthermore, given in Appendix B.

Efficient Computation of $\log\left|\widetilde{C}\right|$ *:*

A practical way to compute $\log|\widetilde{C}|$ without performing an eigenvector expansion is to define the $N \times N$ matrix

$$R \doteq Z^\top Z/N + J_{N,N}/N,$$

where $J_{N,N}$ is the $N \times N$ *all-ones* matrix. $Z^\top Z/N$ shares the $N-1$ nonzero eigenvalues with C and has an additional eigenvalue 0 corresponding to the constant eigenvector $(e_N)_i = 1/\sqrt{N}$. Adding an all-ones matrix preserves all existing eigenvalues while replacing the 0 one with a constant. This leads to the following result:

$$-\frac{1}{2}\log|R| = -\frac{1}{2}\sum_{i=1}^{N-1}\log\lambda_i.$$

Appendix E. Proof of Theorem 1: Fixed Points for a Gaussian Model ($N > d$)

Theorem A1 (1)**.** *If the target density $p(x)$ is a D-dimensional multivariate Gaussian, only $D+1$ particles are needed for Algorithm 2 to converge to the exact target parameters.*

The general fixed-point condition for the dynamics (13) of the position x_i for particle i is given by:

$$(I - \mathbb{E}_{\hat{q}}\left[g(x)(x-m)^\top\right])(x_i - m) - \mathbb{E}_{\hat{q}}[g(x)] = 0.$$

for $i = 1, \ldots, N$. By taking the expectation over all particles, we obtain:

$$\mathbb{E}_{\hat{q}}[g(x)] = 0, \tag{A4}$$

where $\hat{q}$ is the empirical distributions of particles at the the fixed point. Note that this result is independent of N, i.e., it is also valid for $N = 1$.

For a D-dimensional Gaussian target $p(x) = \mathcal{N}(\mu, \Sigma)$, we will show that empirical mean and covariance given by the particle algorithm converge to the true mean and covariance matrix of the Gaussian when we use $N \geq D+1$ particles. In this setting, we have $\varphi(x) = \frac{1}{2}x^\top\Sigma^{-1}x - x^\top\Sigma^{-1}\mu$. For simplification, we use the precision matrix $\Lambda = \Sigma^{-1}$ and get

$$\varphi(x) = \frac{1}{2}x^\top\Lambda x - x^\top\Lambda\mu.$$

The gradient $g(x)$ becomes:

$$g(x) = \Lambda(x - \mu)$$

At the fixed points, we have that $\frac{dm}{dt}$ and $\frac{d\Gamma}{dt}$ are equal to 0. For the mean m:

$$\begin{aligned}\frac{dm}{dt} = \mathbb{E}_{\hat{q}}[g(x)] &= 0\\ \Lambda\mathbb{E}_{\hat{q}}[x-\mu] &= 0\\ \Lambda m &= \Lambda\mu\\ m &= \mu\end{aligned}$$

For the matrix Γ, we have

$$\begin{aligned}\frac{d\Gamma}{dt} = -A\Gamma &= 0\\ \Gamma - \mathbb{E}_{q_0}\left[g(x)(x-m)^\top\right]\Gamma &= 0\\ \mathbb{E}_{q_0}\left[\Lambda(x-\mu)(x-m)^\top\right]\Gamma &= \Gamma\\ -2\eta_2\mathbb{E}_{q_0}\left[(x-m)(x-m)^\top\right]\Gamma &= \Gamma\\ \Lambda C\Gamma &= \Gamma\\ \Lambda C^2 &= C\end{aligned}$$

where we use the result for the mean $m = \mu$ and right multiplied by $\Gamma^\top$ as $C = \Gamma\Gamma^\top$. Now, we can only simplify, as $C = \Lambda^{-1} = \Sigma$ if C is not singular. This is true only if its rank is equal to D, needing $D+1$ particles.

Appendix F. Proof of Theorem 2: Rates of Convergence for Gaussian Targets

Theorem A2 (2). *For a target $p(x) = \mathcal{N}(x \mid \mu, \Lambda^{-1})$, where $x \in \mathbb{R}^D$, and $N \geq D+1$ particles, the continuous time limit of Algorithm 2 will converge exponentially fast for both the mean and the trace of the precision matrix:*

$$\begin{aligned}m^t - \mu &= e^{-\Lambda t}(m^0 - \mu),\\ \mathrm{tr}\left(\left(C^t\right)^{-1} - \Lambda\right) &= e^{-2t}\mathrm{tr}\left(\left(C^0\right)^{-1} - \Lambda\right),\end{aligned}$$

where m^t and C^t are the empirical mean and covariance matrix at time t and $\exp(-\Lambda t)$ is the matrix exponential.

In the following, we assume the target $p(x) = \mathcal{N}(\mu, \Sigma)$ We use the notation $\Lambda \doteq \Sigma^{-1}$ and $\delta C^t = C^t - \Sigma$.

Appendix F.1. Convergence of the Mean

Given our target $p(x)$, similarly to Appendix E we have $g(x) = \Lambda(x-\mu)$, where $\eta_1 = \Sigma^{-1}\mu$ and $\eta_2 = -\frac{1}{2}\Sigma^{-1}$. This transform the first of Equations (11) into

$$\begin{aligned}\frac{dm}{dt} &= -\Lambda(\mathbb{E}_{\hat{q}}[x] - \mu)\\ &= -\Lambda(m-\mu)\end{aligned}$$

If now consider the error on m : $\delta m = m - \mu$ we obtain:

$$\begin{aligned}\frac{d\delta m}{dt} = \frac{dm}{dt} &= -\Lambda(m-\mu)\\ &= -\Lambda\delta m.\end{aligned}$$

Therefore, the mean converges exponentially fast to the true mean. The asymptotic rate is governed by the largest eigenvalue of Λ, i.e., the inverse of the smallest eigenvalue of Σ, $\lambda_{\min}$.

Appendix F.2. Convergence of the Covariance Matrix

Let $z = x - m$, we have from Equation (5), that

$$\frac{dz}{dt} = -Az$$

where $A = \mathbb{E}_{q_0}\left[g(x)z^\top\right] - I$. This expectation can further be simplified as

$$\mathbb{E}_{\hat{q}}\left[\Lambda(x-\mu)z^\top\right] = \Lambda C, \tag{A5}$$

where $q \sim \mathcal{N}(m, C)$. Hence, we have the exact result

$$\frac{dC}{dt} = (I - \Lambda C)C + C(I - C\Lambda). \tag{A6}$$

We know that the optimal target is $C = \Sigma$. Therefore, we define the error $\delta C = C - \Sigma$. Linearizing Equation (A6) gives us

$$\begin{aligned}\frac{d\delta C}{dt} = \frac{dC}{dt} =& (I - \Lambda(\delta C + \Sigma))(\delta C + \Sigma) \\ &+ (\delta C + \Sigma)(I - (\delta C + \Sigma)\Lambda) \\ =& -\Lambda\delta C(\delta C + \Sigma) - (\delta C + \Sigma)\delta C\Lambda \\ \approx& -\Lambda\delta C\Sigma - \Sigma\delta C\Lambda\end{aligned}$$

We were not yet able to find a general solution of this equation, but we can obtain a simple result for the trace $y^t \doteq \mathrm{tr}(\delta C)$ at time t:

$$\frac{dy^t}{dt} \simeq -2y^t.$$

We, therefore, have a asymptotic linear convergence: $y^t \propto e^{-2t}y^0$ which is independent of the parameters of the Gaussian model.

We can also equivalently obtain a non-asymptotic estimate of a specific error measure for the precision matrix. Using equation (A6), we have the following dynamics for the precision C^{-1}:

$$\begin{aligned}\frac{dC^{-1}}{dt} =& -C^{-1}\frac{dC}{dt}C^{-1} \\ =& -C^{-1}(I - \Lambda C) - (I - \Lambda C)C^{-1}\end{aligned}$$

Taking the trace

$$\begin{aligned}\frac{d\mathrm{tr}(C^{-1})}{dt} =& -2\mathrm{tr}(C^{-1}) - 2\mathrm{tr}(\Lambda) \\ \frac{d\mathrm{tr}(C^{-1} - \Lambda)}{dt} =& -2\mathrm{tr}(C^{-1} - \Lambda)\end{aligned}$$

Hence we get the following exact result:

$$\mathrm{tr}((C^t)^{-1} - \Lambda) = e^{-2t}\mathrm{tr}((C^0)^{-1} - \Lambda)$$

which is again independent of the parameters of the Gaussian model.

Additionally, this tells us that if the covariance C is non-singular at time $t = 0$, it will remain non-singular for all t ($\mathrm{tr}(C^{-1})$ would be infinite). Hence, if we start with $N > d$ particles with a proper empirical covariance, they cannot collapse to make C singular.

Appendix F.3. Convergence of the Trace of the Covariance

The asymptotic result on traces obtained previously can be turned into an exact inequality. We have

$$\frac{d\delta C}{dt} = -\Lambda\delta C\Sigma - \Sigma\Lambda\delta C - \Lambda(\delta C)^2 - (\delta C)^2\Lambda$$

Taking the trace, we get

$$\frac{d\mathrm{tr}(\delta C)}{dt} = -2\mathrm{tr}(\delta C) - 2\mathrm{tr}(\delta C\Lambda\delta C)$$

Since $\delta C\Lambda\delta C$ is positive definite, we have $-2\mathrm{tr}(\delta C\Lambda\delta C) \leq 0$ and thus

$$\frac{d\mathrm{tr}(\delta C)}{dt} \leq -2\mathrm{tr}(\delta C)$$

leading to:

$$\mathrm{tr}(\delta C^t) \leq \mathrm{tr}(\delta C^0)e^{-2t}$$

by using by Grönwall's lemma [46]:

Lemma A1 (Grönwall)**.** *For an interval $I_0 = [0, \infty)$ and a given function f differentiable everywhere in I_0 and satisfying:*

$$f'(t) \leq \beta(t)f(t), \quad t \in I_0$$

then f is bounded by the corresponding differential equation $g'(t) = \beta(t)g(t)$:

$$f(t) \leq f(0)\int_0^t \beta(s)ds, \quad t \in I_0$$

The bound is nontrivial only if $\mathrm{tr}(\delta C) \geq 0$. This would be natural assumption for a Bayesian model, if C^0 is the prior covariance and the eigenvalues of C^t at $t = \infty$ (corresponding to the posterior) are reduced by the data.

Appendix F.4. Decay of Fluctuation Part of the Free Energy

Still focusing on the Gaussian model, we can further derive a bound on the free energy. It is easy to see that for the Gaussian case, the free energy in Equation (4) separates into a sum of two terms. The first one depends on the mean m^t only and the second one on only the fluctuations (i.e., C^t).

We will consider the second, nontrivial part only. We assume that the covariance matrix is nonsingular (corresponding to $N > D$). The fluctuation part of the free energy (minus its minimum) is given by

$$\mathcal{F}_{fl} = -\frac{1}{2}\ln|I - B| - \frac{1}{2}\mathrm{tr}(B)$$

where we have introduced the matrix $B \doteq I - \Lambda C$. One can show that its eigenvalues are real and are upper bounded by 1. First, we can show from the equations of motion that

$$\frac{d\mathcal{F}_{fl}}{dt} = -\mathrm{tr}(BB^\top) \tag{A7}$$

Second, using the elementary bound $-\ln(1-u) \leq \frac{u}{1-u}$ valid for $u \leq 1$ and applied to the eigenvalues of B yields

$$\begin{aligned}
\mathcal{F}_{fl} \leq & \frac{1}{2}\mathrm{tr}(B(I-B)^{-1} - B) \\
= & \frac{1}{2}\mathrm{tr}(B(I-B)^{-1} - B(I-B)(I-B)^{-1}) \\
= & \frac{1}{2}\mathrm{tr}(B^2(I-B)^{-1}) \\
= & \frac{1}{2}\mathrm{tr}(B^2C^{-1}\Lambda^{-1}) \leq \frac{1}{2}\mathrm{tr}(B^\top\Lambda^{-1}BC^{-1})
\end{aligned}$$

The last two equalities used the definition $B = I - \Lambda C$. Since $B^\top\Lambda^{-1}B$ and C^{-1} are both positive definite, we can bound the last term by (see ([47], Theorem 6.5))

$$\begin{aligned}
\mathcal{F}_{fl} \leq & \frac{1}{2}\mathrm{tr}(B^\top\Lambda^{-1}B)\mathrm{tr}(C^{-1}) \leq \\
& \frac{1}{2}\mathrm{tr}(BB^\top)\mathrm{tr}(\Lambda^{-1})\mathrm{tr}(C^{-1})),
\end{aligned}$$

where, in the last line, we have bounded the trace of a product of p.d. matrices a second time.

Combining with Equation (A7) we show that

$$\frac{d\mathcal{F}_{fl}}{dt} \leq -\frac{2\mathcal{F}_{fl}}{\mathrm{tr}(\Lambda^{-1})\mathrm{tr}(C^{-1})}$$

We can plug in our result from Theorem 2:

$$\begin{aligned}
\mathrm{tr}(C^{-1}) = & \mathrm{tr}(\Lambda) + \mathrm{tr}(C^{-1} - \Lambda) \\
= & \mathrm{tr}(\Lambda) + e^{-2t}\mathrm{tr}((C^0)^{-1} - \Lambda) \\
\leq & \mathrm{tr}(\Lambda) + e^{-2t}|\mathrm{tr}((C^0)^{-1} - \Lambda)| \\
\leq & \mathrm{tr}(\Lambda) + |\mathrm{tr}((C^0)^{-1} - \Lambda)|
\end{aligned}$$

We can plug this in and use Grönwall's Lemma A1 to get an exponential bound

$$\mathcal{F}_{fl}(C^t) \leq \mathcal{F}_{fl}(C^0)e^{-\left[\frac{2t}{\mathrm{tr}(\Lambda^{-1})(\mathrm{tr}(\Lambda)+|\mathrm{tr}((C^0)^{-1}-\Lambda)|)}\right]}.$$

Appendix F.5. Asymptotic Decay of the Free Energy:

For large times t, we can do better. Let us analyse the asymptotic decay constant $\mathcal{F}_{fl} \simeq e^{-\lambda_{free} t}$ defined by

$$\begin{aligned}\lambda_{free} \doteq -\lim_{t\to\infty} \frac{d\ln(\mathcal{F}_{fl})}{dt} &= -\lim \frac{\frac{d\mathcal{F}_{fl}}{dt}}{\mathcal{F}_{fl}} \\ &= \lim \frac{\mathrm{tr}(BB^\top)}{-\frac{1}{2}\ln|I-B| - \frac{1}{2}\mathrm{tr}(B)} \geq \\ &\lim \frac{\mathrm{tr}(B^2)}{-\frac{1}{2}\ln|I-B| - \frac{1}{2}\mathrm{tr}(B)}\end{aligned}$$

In the last inequality, we used $\mathrm{tr}(BB^\top) \geq \mathrm{tr}(B^2)$. Everything is expressed by traces of functions of B, and thus by its eigenvalues. Since $B \to 0$ as $t \to \infty$ (this applies also to its eigenvalues u), we can use Taylor's expansion $\ln(1-u) + u = -u^2/2 + O(u^3)$ to show that

$$\lambda_{free} \geq 4$$

which is independent of Λ.

Appendix G. Proof of Theorem 3: Fixed-Points for Gaussian Model ($N \leq D$)

Theorem A3 (3). *Given a D-dimensional multivariate Gaussian target density $p(x) = \mathcal{N}(x|\mu, \Sigma)$, using Algorithm 2 with $N < D+1$ particles, the empirical mean converges to the exact mean μ. The $N-1$ non-zero eigenvalues of C^t converge to a subset of the target covariance Σ spectrum. Furthermore, the **global minimum** of the regularised version $\widetilde{\mathcal{F}}$ of the free energy (17) corresponds to the **largest** eigenvalues of Σ.*

Applying Equation (A4) to our fixed point equation, we obtain

$$(I - \mathbb{E}_{\hat{q}}\left[g(x)(x-m)^\top\right])(x_i - m) = 0, \ \forall i = 1, \ldots, N$$

Hence, the set of centered positions of the particles $S = \{x_i - m\}_{i=1}^N$, are all eigenvectors of the matrix $\mathbb{E}_{\hat{q}}\left[g(x)(x-m)^\top\right]$ with eigenvalue 1. S spans a $N-1$ dimensional space (we have $\sum_{i=1}^N (x_i - m) = 0$).

If we specialise to a Gaussian target $p(x) = \mathcal{N}(x \mid \mu, \Sigma)$, (and $\Lambda = \Sigma^{-1}$ we have $g(x) = \Lambda(x-\mu)$ and can reuse the result from Equation (A5):

$$\begin{aligned}\mathbb{E}_{\hat{q}}\left[g(x)(x-m)^\top\right] &= \Lambda \mathbb{E}_{\hat{q}}\left[(x-m)(x-m)^\top\right] \\ &= \Lambda C.\end{aligned}$$

Using the equality above, we get:

$$\begin{aligned}\Lambda C(x_i - m) &= (x_i - m) \\ C(x_i - m) &= \Sigma(x_i - m), \ \forall i = 1, \ldots, N\end{aligned}$$

which shows that the obtained low-rank covariance C and the target covariance Σ have $N-1$ eigenvectors and eigenvalues in common.

However, are these the largest ones? We look at the modified free energy (17) (ignoring the contribution of the mean):

$$\min \widetilde{\mathcal{F}} = \min\left\{ -\frac{1}{2} \sum_{i:\lambda_i>0} \ln \lambda_i + \mathrm{tr}(\Lambda C) \right\}$$

where λ_i are the eigenvalues of the empirical covariance C. We first note that $\mathrm{tr}(\Lambda C) = N-1$, independent of which eigenvalues are obtained at the fixed point. This is easily seen by the following argument: If we use the index–set $\mathcal{I}$ for the common eigenvectors e_i and eigenvalues λ_i, $i \in \mathcal{I}$, we can write

$$C = \sum_{i\in\mathcal{I}} e_i \lambda_i e_i^\top$$
$$\Sigma = \sum_{i} e_i \lambda_i e_i^\top$$

From this we obtain

$$\mathrm{tr}(\Lambda C) = \mathrm{tr}(\sum_{i\in\mathcal{I}} e_i \lambda_i^{-1} \lambda_i e^\top) = N-1$$

From this result we obtain

$$\min \widetilde{\mathcal{F}} = \max \frac{1}{2} \sum_{i:\lambda_i>0} \ln \lambda_i - (N-1),$$

The term $N-1$ is a constant, but the first term makes a difference: The **absolute minimum** of $\widetilde{\mathcal{F}}$ is achieved, when the λ_i are $N-1$ **largest** eigenvalues of Σ. Our simulations empirically show that the algorithm usually converges to the absolute minimum.

Appendix H. Dimension-Wise Optimizers

Here, we list some of the most populars optimizers used and their dimension-wise versions. In all algorithms, we consider φ the matrix created by the concatenation of the flow of each particle : $\varphi = [\varphi_1, \ldots, \varphi_N]$, where $\varphi_n = \varphi(x_n)$ We additionally use the notation $\varphi_{n,i}$ for the i-th dimension of the flow of the n-th particle. The main differences between the original algorithms and their modified version were put in red.

Appendix H.1. ADAM

The ADAM algorithm is given by:

Algorithm A1: ADAM

Input: $\varphi^t, m^{t-1}, v^{t-1}, \beta_1, \beta_2, \eta$
Output: Δ
$m^t_{n,d} = \beta_1 m^{t-1}_{n,d} + (1-\beta_1)\varphi^t_{n,d}$
$v^t_{n,d} = \beta_2 v^{t-1}_{n,d} + (1-\beta_2)\left(\varphi^t_{n,d}\right)^2$
$\Delta_{n,d} = \eta \frac{m^t_{n,d}}{(1-\beta_1^t)\left(\sqrt{v^t_{n,d}(1-\beta_2^t)^{-1}}+\epsilon\right)}$

Algorithm A2: Dimension-wise ADAM

Input: $\varphi^t, m^{t-1}, v^{t-1}, \beta_1, \beta_2, \eta$
Output: Δ
$m^t_{n,d} = \beta_1 m^{t-1}_{n,d} + (1-\beta_1)\varphi^t_{n,d}$;
$v^t_d = \beta_2 v^{t-1}_d + (1-\beta_2)\frac{1}{N}\sum_{n=1}^{N}\left(\varphi^t_{n,d}\right)^2$;
$\Delta_{n,d} = \eta \frac{m^t_{n,d}}{(1-\beta_1^t)\left(\sqrt{v^t_d (1-\beta_2^t)^{-1}}+\epsilon\right)}$;

Appendix H.2. AdaGrad

The AdaGrad algorithm is given by:

Algorithm A3: AdaGrad

Input: φ^t, v^{t-1}, η
Output: Δ
$v^t_{n,d} = v^{t-1}_{n,d} + \left(\varphi^t_{n,d}\right)^2$
$\Delta_{n,d} = \eta \frac{\varphi^t_{n,d}}{\sqrt{v^t_{n,d}}+\epsilon}$

Algorithm A4: Dimension-wise AdaGrad

Input: φ^t, v^{t-1}, η
Output: Δ
$v^t_d = v^{t-1}_d + \frac{1}{N}\sum_{n=1}^{N}\left(\varphi^t_{n,d}\right)^2$
$\Delta_{n,d} = \eta \frac{\varphi^t_{n,d}}{\sqrt{v^t_d}+\epsilon}$

Appendix H.3. RMSProp

The RMSProp algorithm is given by:

Algorithm A5: RMSProp

Input: $\varphi^t, v^{t-1}, \rho, \eta$
Output: Δ
$v^t_{n,d} = \rho v^{t-1}_{n,d} + (1-\rho)\left(\varphi^t_{n,d}\right)^2$
$\Delta_{n,d} = \eta \frac{\varphi^t_{n,d}}{\sqrt{v^t_{n,d}}+\epsilon}$

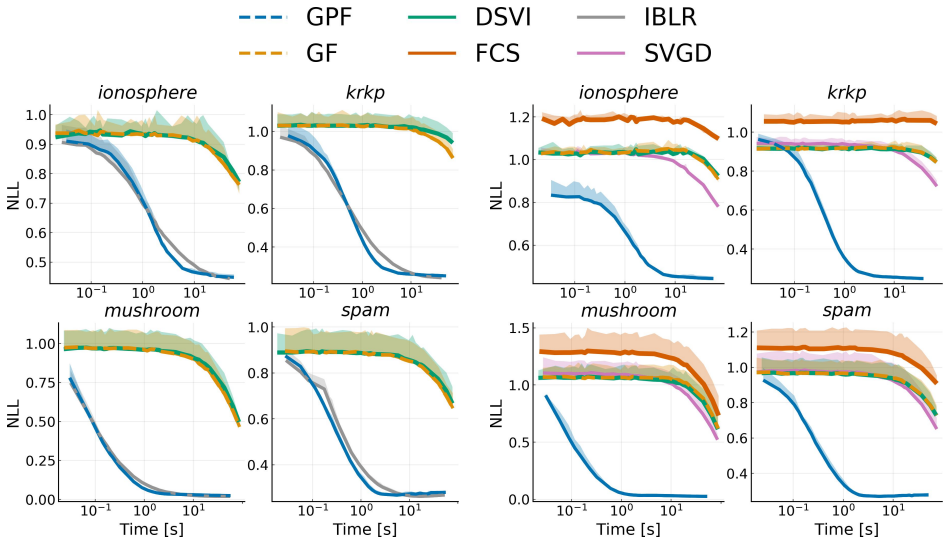
Algorithm A6: Dimension-wise RMSProp

Input: $\varphi^t, v^{t-1}, \rho, \eta$
Output: Δ
$v^t_d = \rho v^{t-1}_d + (1-\rho)\frac{1}{N}\sum_{n=1}^{N}\left(\varphi^t_{n,d}\right)^2$
$\Delta_{n,d} = \eta \frac{\varphi^t_{n,d}}{\sqrt{v^t_d}+\epsilon}$

Appendix I. Additional Figures

Appendix I.1. Bayesian Logistic Regression

Similarly to the previous section, we also show results with the RMSProp optimizer with learning rate 1×10^{-4}.



(**a**) Mean-field approximation (**b**) No mean-field approximation

Figure A1. Similarly to Figure 6, we show the average negative log-likelihood on a test-set over 10 runs against training time on different datasets for a Bayesian logistic regression problem. The dashed curve represents the low-rank approximation with RMSProp for methods based on stochastic estimators.

Appendix I.2. Bayesian Neural Network

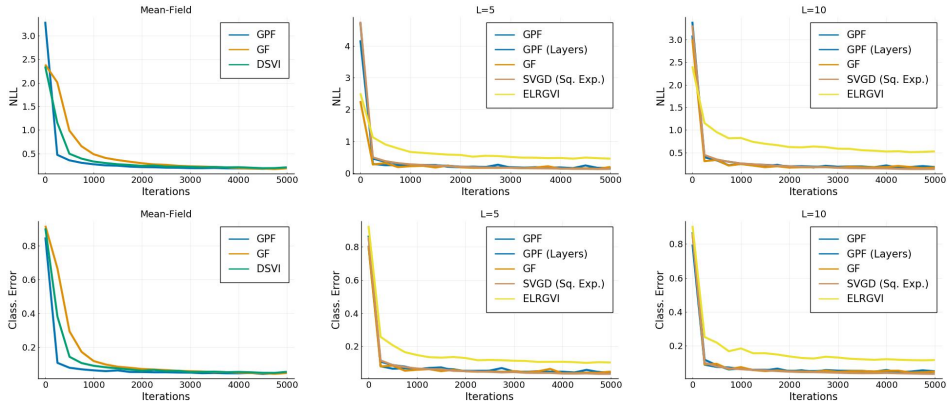


Figure A2. Convergence of the classification error and average negative log-likelihood as a function of time.

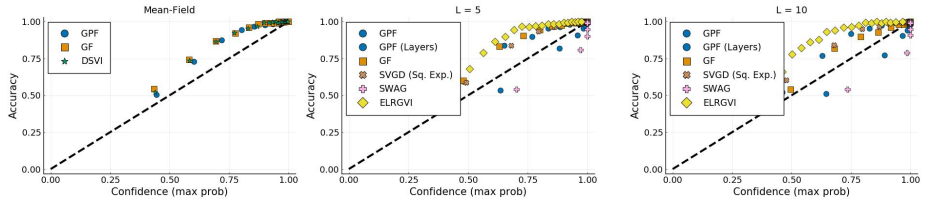


Figure A3. Accuracy vs confidence. Every test sample is clustered in function of its highest predictive probability. The accuracy of this cluster is then computed. A perfectly calibrated estimator would return the identity.

References

1. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2016**, *104*, 148–175.
2. Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; University of Wisconsin–Madison: Madison, WI, USA, 2009.
3. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA, 2018.
4. Bardenet, R.; Doucet, A.; Holmes, C. On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.* **2017**, *18*, 1515–1557.
5. Cowles, M.K.; Carlin, B.P. Markov chain Monte Carlo convergence diagnostics: A comparative review. *J. Am. Stat. Assoc.* **1996**, *91*, 883–904.
6. Barber, D.; Bishop, C.M. Ensemble learning for multi-layer networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1998; pp. 395–401.
7. Graves, A. Practical Variational Inference for Neural Networks. In Proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; Volume 24, pp. 2348–2356.
8. Ranganath, R.; Gerrish, S.; Blei, D. Black box variational inference. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 814–822.
9. Liu, Q.; Lee, J.; Jordan, M. A kernelized Stein discrepancy for goodness-of-fit tests. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 276–284.
10. Liu, Q.; Wang, D. Stein variational gradient descent as moment matching. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 32, pp. 8868–8877
11. Zhuo, J.; Liu, C.; Shi, J.; Zhu, J.; Chen, N.; Zhang, B. Message Passing Stein Variational Gradient Descent. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 6018–6027.
12. Opper, M.; Archambeau, C. The variational Gaussian approximation revisited. *Neural Comput.* **2009**, *21*, 786–792.
13. Challis, E.; Barber, D. Gaussian kullback-leibler approximate inference. *J. Mach. Learn. Res.* **2013**, *14*, 2239–2286.
14. Titsias, M.; Lázaro-Gredilla, M. Doubly stochastic variational Bayes for non-conjugate inference. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1971–1979.
15. Ong, V.M.H.; Nott, D.J.; Smith, M.S. Gaussian variational approximation with a factor covariance structure. *J. Comput. Graph. Stat.* **2018**, *27*, 465–478.
16. Tan, L.S.; Nott, D.J. Gaussian variational approximation with sparse precision matrices. *Stat. Comput.* **2018**, *28*, 259–275.
17. Lin, W.; Schmidt, M.; Khan, M.E. Handling the Positive-Definite Constraint in the Bayesian Learning Rule. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020; Volume 119, pp. 6116–6126.
18. Hinton, G.E.; van Camp, D. Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In Proceedings of the Sixth Annual Conference on Computational Learning Theory, Santa Cruz, CA, USA, 26–28 July 1993; COLT '93; Association for Computing Machinery: New York, NY, USA, 1993; pp. 5–13.
19. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877.
20. Amari, S.I. Natural Gradient Works Efficiently in Learning. *Neural Comput.* **1998**, *10*, 251–276.
21. Khan, M.E.; Nielsen, D. Fast yet simple natural-gradient descent for variational inference in complex models. In Proceedings of the International Symposium on Information Theory and Its Applications (ISITA), Singapore, 28–31 October 2018; pp. 31–35.
22. Lin, W.; Khan, M.E.; Schmidt, M. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 3992–4002.
23. Salimbeni, H.; Eleftheriadis, S.; Hensman, J. Natural Gradients in Practice: Non-Conjugate Variational Inference in Gaussian Process Models. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Lanzarote, Canary Islands, 9–11 April 2018; pp. 689–697.
24. Liu, Q.; Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. *arXiv* **2016**, arXiv:1608.04471.
25. Ba, J.; Erdogdu, M.A.; Ghassemi, M.; Suzuki, T.; Sun, S.; Wu, D.; Zhang, T. Towards Characterizing the High-dimensional Bias of Kernel-based Particle Inference Algorithms. In Proceedings of the 2nd Symposium on Advances in Approximate Bayesian Inference, Vancouver, BC, Canada, 8 December 2019.
26. Tomczak, M.; Swaroop, S.; Turner, R. Efficient Low Rank Gaussian Variational Inference for Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33.
27. Maddox, W.J.; Izmailov, P.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. A simple baseline for bayesian uncertainty in deep learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 13153–13164.
28. Evensen, G. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res. Oceans* **1994**, *99*, 10143–10162.
29. Rezende, D.; Mohamed, S. Variational inference with normalizing flows. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1530–1538.
30. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural ordinary differential equations. In Proceedings of the 32nd International Conference on Neural Information Processing, Montréal, QC, Canada, 3–8 December 2018; pp. 6572–6583.

31. Ingersoll, J.E. *Theory of Financial Decision Making*; Rowman & Littlefield: Lanham, MD, USA, 1987; Volume 3.
32. Barfoot, T.D.; Forbes, J.R.; Yoon, D.J. Exactly sparse gaussian variational inference with application to derivative-free batch nonlinear state estimation. *Int. J. Robot. Res.* **2020**, *39*, 1473–1502.
33. Korba, A.; Salim, A.; Arbel, M.; Luise, G.; Gretton, A. A Non-Asymptotic Analysis for Stein Variational Gradient Descent. In Proceedings of the 32nd International Conference on Neural Information Processing, Virtual, 6–12 December 2020; Volume 33. pp. 4672–4682.
34. Berlinet, A.; Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
35. Zaki, N.; Galy-Fajou, T.; Opper, M. Evidence Estimation by Kullback-Leibler Integration for Flow-Based Methods. In Proceedings of the Third Symposium on Advances in Approximate Bayesian Inference, Virtual Event, January–February 2021.
36. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A fresh approach to numerical computing. *SIAM Rev.* **2017**, *59*, 65–98. doi:10.1137/141000671.
37. Tieleman, T.; Hinton, G. *Lecture 6.5-rmsprop, Coursera: Neural Networks for Machine Learning*; Technical Report; University of Toronto: Toronto, ON, USA, 2012.
38. Zhang, G.; Li, L.; Nado, Z.; Martens, J.; Sachdeva, S.; Dahl, G.; Shallue, C.; Grosse, R.B. Which Algorithmic Choices Matter at Which Batch Sizes? Insights From a Noisy Quadratic Model. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA 2019; Volume 32, pp. 8196–8207.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
40. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: https://archive.ics.uci.edu/ml/datasets.php (accessed on 28 July 2021).
41. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
42. LeCun, Y. The MNIST Database of Handwritten Digits. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 20 July 2021).
43. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
44. Liu, C.; Zhuo, J.; Cheng, P.; Zhang, R.; Zhu, J. Understanding and accelerating particle-based variational inference. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 4082–4092.
45. Zhu, M.H.; Liu, C.; Zhu, J. Variance Reduction and Quasi-Newton for Particle-Based Variational Inference. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020.
46. Gronwall, T.H. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Ann. Math.* **1919**, *20*, 292–296.
47. Zhang, F. *Matrix Theory: Basic Results and Techniques*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.