Tractable Structured Natural-Gradient Descent Using Local Parameterizations

Wu Lin [1] **Frank Nielsen** [2] **Mohammad Emtiyaz Khan** [3] **Mark Schmidt** [1 4]

Abstract

Natural-gradient descent (NGD) on structured parameter spaces (e.g., low-rank covariances) is computationally challenging due to difficult Fisher-matrix computations. We address this issue by using *local-parameter coordinates* to obtain a flexible and efficient NGD method that works well for a wide-variety of structured parameterizations. We show four applications where our method (1) generalizes the exponential natural evolutionary strategy, (2) recovers existing Newton-like algorithms, (3) yields new structured second-order algorithms, and (4) gives new algorithms to learn covariances of Gaussian and Wishart-based distributions. We show results on a range of problems from deep learning, variational inference, and evolution strategies. Our work opens a new direction for scalable structured geometric methods.

1. Introduction

A wide-variety of problems that arise in the field of optimization, inference, and search can be expressed as

$$\min_{q(\mathbf{w})\in\mathcal{Q}} \mathbb{E}_{q(\mathbf{w})}\left[\ell(\mathbf{w})\right] - \gamma\mathcal{H}(q(\mathbf{w})), \qquad (1)$$

where $\mathbf{w}$ is the parameter of interest, $q(\mathbf{w}) \in \mathcal{Q}$ is a distribution, $\mathcal{H}(q(\mathbf{w}))$ is Shannon's entropy, $\ell(\mathbf{w})$ is a loss function, and $\gamma \geq 0$. For example, in problems involving random search (Baba, 1981), stochastic optimization (Spall, 2005), and evolutionary strategies (Beyer, 2001), $q(\mathbf{w})$ is the so-called 'search' distribution used to find a global minimum of a black-box function $\ell(\mathbf{w})$. In reinforcement learning, it can be the policy distribution which minimizes the expected value-function $\ell(\mathbf{w})$ (Sutton et al., 1998), sometimes with entropy regularization (Williams & Peng, 1991; Teboulle, 1992; Mnih et al., 2016). For Bayesian problems, $q(\mathbf{w})$ is the posterior distribution or its approximation and the $\ell(\mathbf{w})$ is the log of the joint distribution (Zellner, 1986) (γ set to 1). Finally, many robust or global optimization techniques employ $q(\mathbf{w})$ to smooth out local minima (Mobahi & Fisher III, 2015; Leordeanu & Hebert, 2008; Hazan et al., 2016), where often $\gamma = 0$. Developing fast and scalable algorithms for solving (1) potentially impacts all these fields.

Natural-gradient descent (NGD) is an attractive algorithm to solve (1) and can speed up the optimization by exploiting the information geometry of $q(\mathbf{w})$ (Wierstra et al., 2008; Sun et al., 2009; Hoffman et al., 2013; Khan & Lin, 2017; Salimbeni et al., 2018). It also unifies a wide-variety of learning algorithms, which can be seen as its instances with a specific $q(\mathbf{w})$ (Khan & Rue, 2020). This includes deep learning (Khan et al., 2018), approximate inference (Khan & Lin, 2017), and optimization (Khan & Rue, 2020; Khan et al., 2017). NGD also has better convergence properties compared to methods that ignore the geometry, for example, Ranganath et al. (2014); Lezcano Casado (2019).

We consider NGD where parameters of $q(\mathbf{w})$ assume special structures, for example, low-rank or sparse Gaussian covariances. For such cases, NGD is often intractable and/or costly due to difficult Fisher Information Matrix (FIM) computations. First, the FIM can be singular for restricted parametrizations (see Fig. 1(I)), which is often addressed with ad-hoc structural approximations, derived on a case-by-case basis (Sun et al., 2013; Akimoto & Hansen, 2016; Li & Zhang, 2017; Mishkin et al., 2018; Tran et al., 2020) (also see Appx. D.4). Second, while we can switch parameterizations, the computation could be ineffecient because the structure might be lost, for example, when switching from sparse precision to covariances. Using automatic differentiation could make the situation worse because such tools are often unaware of the structure (Salimbeni et al., 2018) (also see Appx. G.1). Finally, the choice of parameterizations and approximations themselves involve delicate choices to get a desired computation-accuracy trade-off. For example, for neural networks layer-wise approximations (Sun & Nielsen, 2017; Zhang et al., 2018; Lin et al., 2019a) might be better than low-rank/diagonal structures (Mishkin et al., 2018; Tran et al., 2020; Ros & Hansen, 2008; Khan et al., 2018), but may also involve more computations. Our goal is to address these difficulties and design a flexible method that works well for a variety of structured parameterizations.

We present *local-parameter coordinates* to design flexible and tractable NGD for a variety of structured-parameter

[1] University of British Columbia. [2] Sony Computer Science Laboratories Inc. [3] RIKEN Center for Advanced Intelligence Project. [4] CIFAR AI Chair, Alberta Machine Intelligence Institute. Correspondence to: Wu Lin <yorker.lin@gmail.com >.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

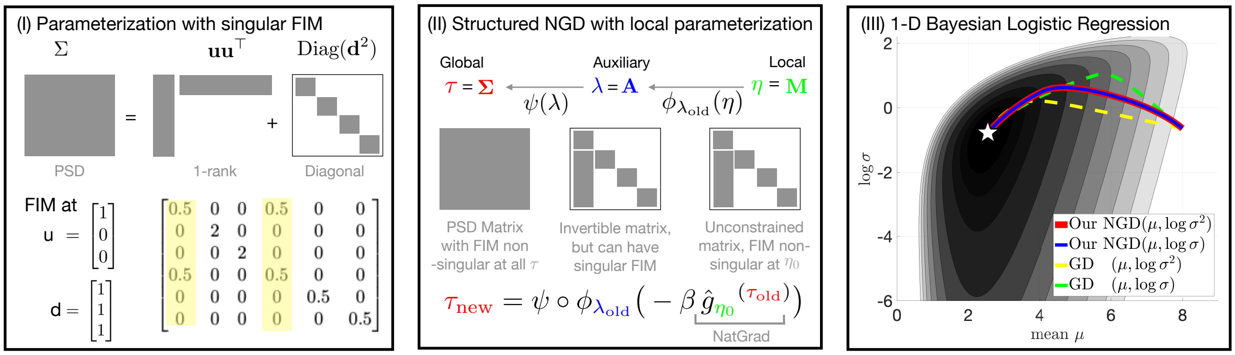

(I) Parameterization with singular FIM
Σ
uu⊤
Diag(d²)
=
+
PSD
1-rank
Diagonal
FIM at
u = [1 0 0]
d = [1 1 1]
0.5 0 0 0.5 0 0
0 2 0 0 0 0
0 0 2 0 0 0
0.5 0 0 0.5 0 0
0 0 0 0 0.5 0
0 0 0 0 0 0.5
(II) Structured NGD with local parameterization
Global
Auxiliary
Local
τ = Σ
ψ(λ)
λ = A
φλold(η)
η = M
PSD Matrix with FIM non-singular at all τ
Invertible matrix, but can have singular FIM
Unconstrained matrix, FIM non-singular at η0
τnew = ψ ∘ φλold(−β ĝη0(τold))
NatGrad
(III) 1-D Bayesian Logistic Regression
2
0
-2
-4
-6
log σ
0
2
4
6
8
mean μ
Our NGD(μ, log σ²)
Our NGD(μ, log σ)
GD (μ, log σ²)
GD (μ, log σ)

Figure 1. (I) The FIM can be singular, for example, when the covariance $\mathbf{\Sigma}$ has a low rank structure (more details in Appx. J.1.6). The two identical columns of FIM are shown in yellow. (II) We fix such issues by using a local parameterization $\boldsymbol{\eta}$ (here $\mathbf{M}$, an unconstrained structured matrix) which is related to the global variable $\boldsymbol{\tau}$ ($= \mathbf{\Sigma}$ for the low-rank example) through an auxiliary parameter $\boldsymbol{\lambda}$ ($= \mathbf{A}$, an invertible matrix with a specific structure to get a low-rank $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\top$). The three parameter-spaces are related through maps $\boldsymbol{\tau} = \boldsymbol{\psi}(\boldsymbol{\lambda}) = \mathbf{A}\mathbf{A}^\top$ and $\boldsymbol{\lambda} = \boldsymbol{\phi}_{\boldsymbol{\lambda}_{\text{old}}}(\boldsymbol{\eta}) = \mathbf{A} = \mathbf{A}_{\text{old}}\text{Exp}(\mathbf{M})$, and need to satisfy Assumptions 1 and 2 given in Section 3. This results in a valid NGD step (shown at the bottom) in the local-parameter space (defined at $\boldsymbol{\eta}_0 = 0$ with learning rate β). (III) For a 1-D Bayesian logistic-regression, our NGD is invariant to two different parameterizations, which is not the case for GD (details in Appx. D.3).

spaces. The method is summarized in Fig. 1(II), and involves specifying (i) a 'local parameter coordinate' that satisfies the structural constraints of the original (global) parameters, (ii) a map to convert back to the global parameters via 'auxiliary' parameters, and finally (iii) a tractable natural-gradient computation in the local-parameter space. This construction ensures a valid NGD update in local parameter spaces, while maintaining structures (often via matrix groups) in the auxiliary parameters. This decoupling enables a tractable NGD that exploits the structure, when these parameters and the map are chosen carefully.

We show four applications of our method.

1. We generalize Glasmachers et al. (2010)'s method to more general distributions and structures (Section 3.1).

2. In Section 3.2, we recover Newton-like methods derived by Lin et al. (2020) using Riemannian-gradients and by Khan et al. (2018) using the standard NGD.

3. Our approach is easily generalizable to other non-Gaussian cases; see Setion 3.3 and 3.4.

4. In Section 4, we derive new 2$^{\text{nd}}$-order methods for low-rank, diagonal, and sparse covariances. The methods are only slightly more costly than diagonal-covariance methods. Moreover, they can be used as structured 2$^{\text{nd}}$-order methods for unconstrained optimization.

We show applications to various problems for search, variational inference, and deep learning, obtaining much faster convergence than methods that ignore geometry. An example for 1-D logistic regression is shown in 1(III). Overall, our work opens a new direction to design efficient and structured geometric methods via local parameterizations.

2. Structured NGD and its Challenges

The distributions $q(\mathbf{w}) \in \mathcal{Q}$ are often parameterized, say using parameters $\boldsymbol{\tau} \in \Omega_\tau$, for which we write $q(\mathbf{w}|\boldsymbol{\tau})$. The problem can then be conveniently expressed as an optimization problem in the space Ω_τ,

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}\in\Omega_\tau} \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau})}\left[\ell(\mathbf{w})\right], \tag{2}$$

where we assume $\gamma = 0$ for simplicity (general case is in Lemma 4 of Appx. C). The NGD step is $\boldsymbol{\tau}_{t+1} \leftarrow \boldsymbol{\tau}_t - \beta\hat{\boldsymbol{g}}_{\boldsymbol{\tau}_t}$ where $\beta > 0$ is the step size and natural gradients are as

$$\hat{\boldsymbol{g}}_{\boldsymbol{\tau}_t} := \mathbf{F}_{\boldsymbol{\tau}}(\boldsymbol{\tau}_t)^{-1}\nabla_{\boldsymbol{\tau}}\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau})}\left[\ell(\mathbf{w})\right], \tag{3}$$

where $\mathbf{F}_{\boldsymbol{\tau}}(\boldsymbol{\tau}) := \mathbb{E}_q[\nabla_\tau \log q(\mathbf{w}|\boldsymbol{\tau})(\nabla_\tau^\top \log q(\mathbf{w}|\boldsymbol{\tau}))]$ is an invertible and well-defined FIM following the regularity condition (see Appx. C). The iterates $\boldsymbol{\tau}_{t+1}$ may not always lie inside Ω_τ and a projection step might be required.

In some cases, the NGD computation may not require an explicit FIM inversion. For example, when $q(\mathbf{w}|\boldsymbol{\tau})$ is a minimal exponential-family (EF) distribution, FIM is always invertible, and natural gradients are equal to vanilla gradients with respect to the 'expectation parameter' (Malagò et al., 2011; Khan & Nielsen, 2018). By appropriately choosing $\mathcal{Q}$, the NGD then takes forms adapted by popular algorithms (Khan & Rue, 2020), for example, for Gaussians $q(\mathbf{w}|\boldsymbol{\tau}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \mathbf{S}^{-1})$ where $\mathbf{S}$ denotes the precision, it reduces to a Newton-like update (Khan et al., 2018),

$$\begin{aligned}\boldsymbol{\mu}_{t+1} &\leftarrow \boldsymbol{\mu}_t - \beta\mathbf{S}_{t+1}^{-1}\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau}_t)}[\nabla_w \ell(\mathbf{w})], \\ \mathbf{S}_{t+1} &\leftarrow \mathbf{S}_t + \beta\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau}_t)}\left[\nabla_w^2 \ell(\mathbf{w})\right].\end{aligned} \tag{4}$$

The standard Newton update for optimization is recovered by approximating the expectation at the mean and using a step-size of 1 with $\gamma = 1$ (Khan & Rue, 2020). Several connections and extensions have been derived in the recent years establishing NGD as an important algorithm for optimization, search, and inference (Khan & Lin, 2017; Khan & Nielsen, 2018; Lin et al., 2019a; Osawa et al., 2019b).

This simplification of NGD breaks down when (2) involves structured-parameter spaces Ω_τ, for example, spaces with constrains such as low-rank or sparse structures. Even for the simplest Gaussian case, where covariances lie in the positive-definite space, the update (4) may violate the constraint (Khan et al., 2018). Extensions have been derived using Riemannian gradient descent (RGD) to fix this issue (Lin et al., 2020). Other solutions based on Cholesky (Sun et al., 2009; Salimbeni et al., 2018) or square-root parameterization (Glasmachers et al., 2010) have also been considered, where the problem is converted to an unconstrained parameter space. For example, Glasmachers et al. (2010) use a square-root parameterization $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T)$, where $\mathbf{A}$ is the square-root of $\mathbf{S}^{-1}$, to get the update,

$$\begin{aligned}\boldsymbol{\mu}_{t+1} &\leftarrow \boldsymbol{\mu}_t - \beta \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau}_t)}\left[(\mathbf{A}_t \mathbf{z}_t)\ell(\mathbf{w})\right],\\ \mathbf{A}_{t+1} &\leftarrow \mathbf{A}_t \mathrm{Exp}\left(-\frac{\beta}{2}\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau}_t)}\left[(\mathbf{z}_t\mathbf{z}_t^T - \mathbf{I})\ell(\mathbf{w})\right]\right), \quad (5)\end{aligned}$$

where $\mathbf{z}_t = \mathbf{A}_t^{-1}(\mathbf{w} - \boldsymbol{\mu}_t)$ and $\mathrm{Exp}(\mathbf{X}) = \mathbf{I} + \sum_{k=1}^\infty \frac{\mathbf{X}^k}{k!}$ is the matrix exponential function. These solutions however do not easily generalize. For example, it is not obvious how to apply these updates to cases where the covariance is low-rank (Mishkin et al., 2018; Tran et al., 2020), Kronecker structured (Zhang et al., 2018; Lin et al., 2019a), or to cases involving non-Gaussian distributions such as the Wishart, univariate exponential family distributions (Lin et al., 2020) and Gaussian mixtures (Lin et al., 2019a).

In fact, the issue with the structure and its effect on parameterization is a bit more involved than it might appear at first. Certain choices of the structure/parameterization can make the Fisher matrix singular which can make NGD invalid, for example, for low-rank Gaussians as shown in Fig. 1(I) where it requires new tricks such as auxiliary parameterization (Lin et al., 2019a), block approximations (Tran et al., 2020), algorithmic approximations (Mishkin et al., 2018), or damping (Zhang et al., 2018). The computational cost depends on the parameterization, the choice of which is often not obvious. Some methods exploit structure in the covariances (Glasmachers et al., 2010) while the others work with its inverse such as (4). Customized structures, such as layer-wise and Kronecker-factored covariances in deep neural nets, may work well in one parameterization but not in the other. Thus, it is essential to have a flexible method that works well for a variety of structured-parameterizations and distributions. Our goal is to propose such a method.

3. Local Parameter Coordinates

We present local-parameter coordinates to obtain a flexible and efficient NGD method that works well for a wide-variety of structured parameterizations. Table 1 in Appx. A summarizes the examples and extensions we consider. We describe the method in three steps.

Step 1. The first step involves specifying a 'local' parameterization, denoted by $\boldsymbol{\eta} \in \Omega_\eta$, so that the following assumption is satisfied (throughout, we set $\boldsymbol{\eta}_0 = \mathbf{0}$).

Assumption 1: *The Fisher matrix* $\mathbf{F}_{\boldsymbol{\eta}}(\boldsymbol{\eta}_0)$ *is non-singular.*

Step 2. The second step involves specifying two maps shown below to connect to the original 'global' parameters $\boldsymbol{\tau}$ via an 'auxiliary' parameter $\boldsymbol{\lambda} \in \Omega_\lambda$,

$$\boldsymbol{\tau} = \boldsymbol{\psi}(\boldsymbol{\lambda}) \text{ and } \boldsymbol{\lambda} = \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta}), \tag{6}$$

where the first map is surjective and the second map is defined such that $\boldsymbol{\lambda}_t = \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta}_0)$, i.e., the function is tight at $\boldsymbol{\eta}_0$ to match the current $\boldsymbol{\lambda}_t$. The local parameter $\boldsymbol{\eta}_0$ can be seen as a *relative origin* tied to $\boldsymbol{\lambda}_t$. The overall map is $\boldsymbol{\tau} = \boldsymbol{\psi} \circ \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})$ (the map could change with iterations). Notice that we make no assumption about the non-singularity of the FIM in the auxiliary space Ω_λ. The FIM in the auxiliary space Ω_λ can be singular (see Section 3.1). The only restriction is a mild *coordinate compatibility* assumption.

Assumption 2 : $\forall \boldsymbol{\lambda}_t \in \Omega_\lambda$, *the map* $\boldsymbol{\eta} \mapsto \boldsymbol{\psi} \circ \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})$ *is locally* C^1*-diffeomorphic at an open neighborhood of* $\boldsymbol{\eta}_0$.

Assumption 2 implies that the local $\boldsymbol{\eta}$ has the same degrees of freedom as $\boldsymbol{\tau}$, but the auxiliary $\boldsymbol{\lambda}$ can have a different one (an example is in Section 3.1). Assumption 1-2, together with surjective $\boldsymbol{\psi}(\cdot)$, imply a non-singular FIM in the global space Ω_τ, so there is no need to check it for specific cases. On the other hand, if we know the non-singularity of the FIM in Ω_τ beforehand, Assumption 2 together with surjective $\boldsymbol{\psi}(\cdot)$ imply that Assumption 1 is satisfied.

Step 3. The final step is to compute the natural gradient at $\boldsymbol{\eta}_0$ in the local-parameter space to update the global $\boldsymbol{\tau}$, which can be done by using the chain rule,

$$\hat{\boldsymbol{g}}_{\boldsymbol{\eta}_0}^{(t)} = \mathbf{F}_{\boldsymbol{\eta}}(\boldsymbol{\eta}_0)^{-1} \nabla_{\boldsymbol{\eta}_0}\left[\boldsymbol{\psi} \circ \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})\right] \mathbf{g}_{\boldsymbol{\tau}_t}, \tag{7}$$

where $\mathbf{g}_{\boldsymbol{\tau}} := \nabla_{\boldsymbol{\tau}} \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau})}[\ell(\mathbf{w})]$ is the vanilla gradient. An indirect computation is given in (26) in Appx. C. The above computation is most useful when the computation of $\hat{\boldsymbol{g}}_{\boldsymbol{\eta}_0}^{(t)}$ is tractable, which ultimately depends on the choice of $\boldsymbol{\psi} \circ \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}$ which in turn depends on the form of $q(\mathbf{w})$. Then, by using an NGD step $\boldsymbol{\eta}_0 - \beta \hat{\boldsymbol{g}}_{\boldsymbol{\eta}_0}^{(t)}$ in the local-parameter space, we get the following overall update for $\boldsymbol{\tau}$,

Structured NGD using local parameters

$$\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}\left(-\beta \hat{\boldsymbol{g}}_{\boldsymbol{\eta}_0}^{(t)}\right), \quad \boldsymbol{\tau}_{t+1} \leftarrow \boldsymbol{\psi}\left(\boldsymbol{\lambda}_{t+1}\right) \tag{8}$$

since we assume $\boldsymbol{\eta}_0 = \mathbf{0}$. In summary, given an auxiliary parameter $\boldsymbol{\lambda}_t$, we can use the natural gradient $\hat{\boldsymbol{g}}_{\boldsymbol{\eta}_0}^{(t)}$ to update $\boldsymbol{\tau}$ according to (8). The NGD step using (3) is a special case of the above NGD step (see details in Appx. F).

Finally, we require the following Assumption to be satisfied to ensure that the NGD step $-\beta \hat{\boldsymbol{g}}^{(t)}_{\boldsymbol{\eta}_0} \in \Omega_\eta$ in (8) (this assumption is satisfied for all examples we discuss).

Assumption 3 : *Ω_η has a vector space structure so that the vector addition, the vector subtraction, and the real scalar multiplication are valid.*

We will now discuss three applications of our method where we derive existing NGD strategies as special cases.

3.1. Gaussian with square-root covariance structure

For a Gaussian distribution $\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the covariance matrix $\boldsymbol{\Sigma}$ is positive definite. Standard NGD steps such as (4), may violate this constraint (Khan et al., 2018). Glasmachers et al. (2010) use $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\top$ where $\mathbf{A}$ is an invertible matrix (not a Cholesky), and derive an update using a specific local parameterization. We now show that their update is a special case of our method.

Following Glasmachers et al. (2010), we choose the following parameterizations, where we use $\mathcal{S}^{p\times p}_{++}$, $\mathcal{S}^{p\times p}$, and $\mathcal{R}^{p\times p}_{++}$ to denote the set of symmetric positive definite matrices, symmetric matrices, and invertible matrices, respectively,

$$\begin{aligned} \boldsymbol{\tau} &:= \left\{ \boldsymbol{\mu} \in \mathbb{R}^p, \;\; \boldsymbol{\Sigma} \in \mathcal{S}^{p\times p}_{++} \right\}, \\ \boldsymbol{\lambda} &:= \left\{ \boldsymbol{\mu} \in \mathbb{R}^p, \;\; \mathbf{A} \in \mathcal{R}^{p\times p}_{++} \right\}, \\ \boldsymbol{\eta} &:= \left\{ \boldsymbol{\delta} \in \mathbb{R}^p, \;\; \mathbf{M} \in \mathcal{S}^{p\times p} \right\}, \end{aligned} \tag{9}$$

where $\boldsymbol{\delta}$ and $\mathbf{M}$ are the local parameters. The map $\boldsymbol{\psi} \circ \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})$ at $\boldsymbol{\lambda}_t := \{\boldsymbol{\mu}_t, \mathbf{A}_t\}$ is chosen to be[1]

$$\begin{aligned} \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \boldsymbol{\Sigma} \end{array} \right\} &= \boldsymbol{\psi}(\boldsymbol{\lambda}) := \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{A}\mathbf{A}^\top \end{array} \right\} \\ \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{A} \end{array} \right\} &= \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta}) := \left\{ \begin{array}{c} \boldsymbol{\mu}_t + \mathbf{A}_t \boldsymbol{\delta} \\ \mathbf{A}_t \mathrm{Exp}\left(\frac{1}{2}\mathbf{M}\right) \end{array} \right\}. \end{aligned} \tag{10}$$

Finally, we can get the natural gradients (7) by using the Fisher matrix $\mathbf{F}_{\boldsymbol{\eta}}(\boldsymbol{\eta}_0)$ (see Appx. D.2 for a derivation),

$$\begin{pmatrix} \hat{\boldsymbol{g}}^{(t)}_{\boldsymbol{\delta}_0} \\ \mathrm{vec}(\hat{\boldsymbol{g}}^{(t)}_{\mathbf{M}_0}) \end{pmatrix} = \begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & \frac{1}{2}\mathbf{I}_{p^2} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{A}_t^\top \mathbf{g}_{\boldsymbol{\mu}_t} \\ \mathrm{vec}(\mathbf{A}_t^\top \mathbf{g}_{\boldsymbol{\Sigma}_t} \mathbf{A}_t) \end{pmatrix} \tag{11}$$

By plugging (10) and (11) in (7), our update can be written in the space of $\boldsymbol{\lambda}$ as below, where $\mathbf{S}_t^{-1} = \boldsymbol{\Sigma}_t$.

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &\leftarrow \boldsymbol{\mu}_t - \beta \mathbf{S}_t^{-1} \mathbf{g}_{\mu_t} \\ \mathbf{A}_{t+1} &\leftarrow \mathbf{A}_t \mathrm{Exp}\big(-\beta \mathbf{A}_t^T \mathbf{g}_{\Sigma_t} \mathbf{A}_t \big) \end{aligned} \tag{12}$$

By the REINFORCE trick (Williams, 1992), the gradients with respect to global parameters are

$$\begin{aligned} \mathbf{g}_\mu &= \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau})}\Big[\big(\mathbf{A}^{-T}\mathbf{z}\big)\ell(\mathbf{w}) \Big] \\ \mathbf{g}_\Sigma &= \tfrac{1}{2}\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\tau})}\Big[\mathbf{A}^{-T}\big(\mathbf{z}\mathbf{z}^T - \mathbf{I}\big)\mathbf{A}^{-1}\ell(\mathbf{w}) \Big] \end{aligned} \tag{13}$$

where $\mathbf{z} = \mathbf{A}^{-1}(\mathbf{w} - \boldsymbol{\mu})$. By plugging in (13) into (12), we recover the update (5) used in Glasmachers et al. (2010). Appx. D.2 shows that Assumptions 1-2 are satisfied.

Parameterizations $\boldsymbol{\eta} = \{\boldsymbol{\delta}, \mathbf{M}\}$ and $\boldsymbol{\lambda} = \{\boldsymbol{\mu}, \mathbf{A}\}$ play distinct roles. Local parameter $\mathbf{M}$ is chosen to be symmetric with $p(p+1)/2$ degrees of freedom so that Assumption 1 holds (also see Appx. D.1.3). Auxiliary parameter $\mathbf{A}$ can be an invertible matrix with p^2 degrees of freedom and the Fisher matrix $\mathbf{F}_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$ is singular. Note that we perform natural-gradient descent in $\boldsymbol{\eta}$ instead of $\boldsymbol{\lambda}$. This is in contrast with the other works (Sun et al., 2009; Salimbeni et al., 2018) that require a Cholesky structure in $\mathbf{A}$ with $p(p+1)/2$ degrees of freedom to ensure that $\mathbf{F}_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$ is non-singular.

Glasmachers et al. (2010) only demonstrated their method in the Gaussian case without complete derivations[2] and a formal formulation. It is difficult to generalize their method without explicitly knowing the distinct roles of parameterizations $\boldsymbol{\eta}$ and $\boldsymbol{\lambda}$. Moreover, their approach only applied to a square-root structure of the covariance and it is unclear how to generalize it to other structures (e.g., low-rank structures). Our method fixes these issues of their approach.

3.2. Connection to Newton's method

We now show that the update (5) derived using local parameterization is in fact closely related to a Newton-like algorithm. Specifically, we will convert the update of $\mathbf{A}_{t+1}$ in (5) to the update over $\mathbf{S}_{t+1}$, as in (4), and recover the Newton's update derived by Lin et al. (2020). To do so, we need to make two changes. First, we will expand $\mathrm{Exp}\big(\beta\mathbf{M}\big) =$

$$\mathbf{I} + \sum_{k=1}^{\infty} \frac{(\beta\mathbf{M})^k}{k!} = \mathbf{I} + \beta\mathbf{M} + \tfrac{1}{2}(\beta\mathbf{M})^2 + O(\beta^3). \tag{14}$$

Second, instead of using (13), we will use Stein's identity (Opper & Archambeau, 2009; Lin et al., 2019b):

$$\mathbf{g}_\mu = \mathbb{E}_q[\nabla_w \ell(\mathbf{w})], \quad \mathbf{g}_\Sigma = \tfrac{1}{2}\mathbb{E}_q\big[\nabla_w^2 \ell(\mathbf{w})\big] \tag{15}$$

Using these changes, the update over $\mathbf{S}_{t+1}$ can be rewritten as a modified Newton's update proposed by Lin et al. (2020),

$$\begin{aligned} \mathbf{S}_{t+1} &= \big(\mathbf{A}_{t+1}\mathbf{A}_{t+1}^T\big)^{-1} = \mathbf{A}_t^{-T}\mathrm{Exp}\big(2\beta\mathbf{A}_t^T\mathbf{g}_\Sigma\mathbf{A}_t\big)\mathbf{A}_t^{-1} \\ &= \mathbf{S}_t + \beta\mathbb{E}_q\left[\nabla_w^2\ell(\mathbf{w})\right] + \frac{\beta^2}{2}\mathbf{G}\mathbf{S}_t^{-1}\mathbf{G} + O(\beta^3) \end{aligned} \tag{16}$$

where $\mathbf{G} = \mathbb{E}_q\left[\nabla_w^2\ell(\mathbf{w})\right]$. Ignoring the red term gives us the update (4) derived by Khan et al. (2018). The term is added by Lin et al. (2020) to fix the positive-definite constraint violation, by Riemannian gradient descent. Thus, these methods can be seen as special cases of ours with an approximation of the exponential map.

[1] The 1/2 shown in red in (10) is used to match the parameterizations in Glasmachers et al. (2010), but the update in (12) remains unchanged even when without it.

[2] There are a few typos in their paper. The matrix $\mathbf{A}$ is missing in their Eq 8 and a factor 2 is missing in Eq 11.

3.3. Wishart with square-root precision structure

We will now show an example that goes beyond Gaussians. We consider a Wishart distribution which is a distribution over p-by-p positive-definite matrices,

$$\mathcal{W}_p(\mathbf{W}|\mathbf{S}, n) = \frac{|\mathbf{W}|^{(n-p-1)/2}|\mathbf{S}|^{n/2}}{\Gamma_p(\frac{n}{2})2^{np/2}} e^{-\frac{1}{2}\mathrm{Tr}(\mathbf{SW})},$$

where $\Gamma_p(\cdot)$ is the multivariate gamma function. Here, the global parameters are based on the precision matrix $\mathbf{S}$, unlike the example in Sec. 3.1. We will see that our update will automatically take care of this difference and report a similar update to the one obtained using $\mathbf{\Sigma}$ in (12).

We start by specifying the parameterization,

$$\begin{aligned}
\boldsymbol{\tau} &:= \left\{ n \in \mathbb{R},\ \ \mathbf{S} \in \mathcal{S}_{++}^{p\times p} \mid n > p-1 \right\}, \\
\boldsymbol{\lambda} &:= \left\{ b \in \mathbb{R},\ \ \mathbf{B} \in \mathcal{R}_{++}^{p\times p} \right\}, \\
\boldsymbol{\eta} &:= \left\{ \delta \in \mathbb{R},\ \ \mathbf{M} \in \mathcal{S}^{p\times p} \right\},
\end{aligned}$$

and their respective maps defined at $\boldsymbol{\lambda}_t := \{b_t, \mathbf{B}_t\}$

$$\begin{aligned}
\left\{ \begin{array}{c} n \\ \mathbf{S} \end{array} \right\} &= \boldsymbol{\psi}(\boldsymbol{\lambda}) := \left\{ \begin{array}{c} 2f(b)+p-1 \\ (2f(b)+p-1)\mathbf{B}\mathbf{B}^\top \end{array} \right\}, \\
\left\{ \begin{array}{c} b \\ \mathbf{B} \end{array} \right\} &= \boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta}) := \left\{ \begin{array}{c} b_t+\delta \\ \mathbf{B}_t\mathrm{Exp}\left(\mathbf{M}\right) \end{array} \right\}.
\end{aligned}$$

where $f(b) = \log(1+\exp(b))$ is the soft-plus function[3]. The auxiliary parameter $\mathbf{B}$ here is defined as the square-root of the *precision* matrix $\mathbf{S}$, unlike in the previous examples.

Denoting the gradients by

$$\mathbf{G}_{\mathbf{S}^{-1}} := \nabla_{\mathbf{S}^{-1}}\mathbb{E}_q\left[\ell(\mathbf{W})\right], \quad g_n := \nabla_n \mathbb{E}_q\left[\ell(\mathbf{W})\right], \quad (17)$$

we can write the updates as (derivation in Appx. E):

$$\mathbf{B}_{t+1} \leftarrow \mathbf{B}_t\mathrm{Exp}\left(\frac{\beta}{n_t^2}\mathbf{B}_t^{-1}\mathbf{G}_{\mathbf{S}_t^{-1}}\mathbf{B}_t^{-T}\right) \qquad (18)$$

$$b_{t+1} \leftarrow b_t - \beta c_t\left[g_n - \frac{1}{n_t}\mathrm{Tr}\left(\mathbf{G}_{\mathbf{S}_t^{-1}}\mathbf{S}_t^{-1}\right)\right] \qquad (19)$$

where $c_t = \frac{2(1+\exp(b_t))}{\exp(b_t)}\left(-\frac{2p}{n_t}+D_{\psi,p}(\frac{n_t}{2})\right)^{-1}$ and $D_{\psi,p}(x)$ is the multivariate trigamma function. Moreover, we can use *re-parameterizable* gradients (Figurnov et al., 2018; Lin et al., 2019b) for $\mathbf{G}_{\mathbf{S}_t^{-1}}$ and g_n due to the Bartlett decomposition (Smith et al., 1972) (see Appx. E.1 for details).

The update (18) for $\mathbf{B}$ (square-root of the precision matrix) is very similar to the update for $\mathbf{A}$ (square-root for covariance) in (12). The change from covariance to precision parameterization changes the sign of the update. The step size is modified using the parameter n_t. The local parameterization can automatically adjust to such changes in the parameter specification, giving rise to intuitive updates.

3.4. Connection to Riemannian Gradient Descent

We will show that the updates on the Wishart distribution is a generalization of Riemannian Gradient Descent (RGD) over the space of positive-definite matrices. Given an optimization problem

$$\min_{Z\in\mathcal{S}_{++}^{p\times p}} \ell(\mathbf{Z})$$

over the space of symmetric positive-definite matrices, the RGD update with retraction can be written in terms of the inverse $\mathbf{U} = \mathbf{Z}^{-1}$ (see Appx. E.2 for the details),

$$\mathbf{U}_{t+1} \leftarrow \mathbf{U}_t + \beta_1\nabla\ell(\mathbf{Z}_t) + \frac{\beta_1^2}{2}\left[\nabla\ell(\mathbf{Z}_t)\right]\mathbf{U}_t^{-1}\left[\nabla\ell(\mathbf{Z}_t)\right]$$

where ∇ is taken with respect to $\mathbf{Z}$, and β_1 is the step size. We now show that this is a special case of (18) where gradients (17) are approximated at the mean of the Wishart distribution as $\mathbb{E}_q\left[\mathbf{W}\right] = n\mathbf{S}^{-1}$. Denoting the mean by $\mathbf{Z}_t$, the approximation is (see the derivation in Appx. E.3),

$$\mathbf{G}_{\mathbf{S}_t^{-1}} \approx n_t\nabla\ell(\mathbf{Z}_t), \quad g_{n_t} \approx \mathrm{Tr}\left[\nabla\ell(\mathbf{Z}_t)\mathbf{S}_t^{-1}\right] \qquad (20)$$

Plugging (20) into (19), b remains constant after the update,

$$b_{t+1} \leftarrow b_t - \beta c_t\left[\cancel{\mathrm{Tr}\left[\nabla\ell(\mathbf{Z}_t)\mathbf{S}_t^{-1}\right]} - \cancel{\mathrm{Tr}\left[\nabla\ell(\mathbf{Z}_t)\mathbf{S}_t^{-1}\right]}\right]$$

so that $b_{t+1} \leftarrow b_t$ and n_t is constant since $n = 2f(b)+p-1$. Resetting the step-size to be $\beta = \frac{1}{2}\beta_1 n$,[4] (18) becomes

$$\mathbf{B}_{t+1} \leftarrow \mathbf{B}_t\mathrm{Exp}\left(\frac{\beta_1}{2}\mathbf{B}_t^{-1}\left[\nabla\ell(\mathbf{Z}_t)\right]\mathbf{B}_t^{-T}\right) \qquad (21)$$

Finally, we express the update in terms of $\mathbf{U}_t := \mathbf{Z}_t^{-1} = \mathbf{B}_t\mathbf{B}_t^T$ to rewrite (21) as by using the second-order terms in the matrix exponential (14),

$$\begin{aligned}
&\mathbf{U}_{t+1} \leftarrow \mathbf{B}_t\mathrm{Exp}(\beta_1\mathbf{B}_t^{-1}\left[\nabla\ell(\mathbf{Z}_t)\right]\mathbf{B}_t^{-T})\mathbf{B}_t^T \\
&\leftarrow \mathbf{U}_t + \beta_1\nabla\ell(\mathbf{Z}_t) + \frac{\beta_1^2}{2}\left[\nabla\ell(\mathbf{Z}_t)\right]\mathbf{U}_t^{-1}\left[\nabla\ell(\mathbf{Z}_t)\right] + O(\beta_1^3)
\end{aligned}$$

recovering the RGD update. Thus, the RGD update is a special case of our update, where the expectation is approximated at the mean. This is a *local* approximation to avoid sampling from $q(\mathbf{W})$. This derivation is another instance of reduction to a *local* method using NGD over distributions, similar to the ones obtained by Khan & Rue (2020).

3.5. Generalizations and Extensions

In previous sections, we use the matrix exponential map to define $\boldsymbol{\phi}_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})$, but other maps can be used. This is convenient since the map can be difficult to compute and numerically unstable. We propose to use another map:

$$\mathbf{h}(\mathbf{M}) = \mathbf{I} + \mathbf{M} + \tfrac{1}{2}\mathbf{M}^2.$$

Map $\mathbf{h}(\cdot)$ plays a key role for complexity reduction in Sec. 4, since it simplifies the natural-gradient computation in Gaussian and Wishart cases without changing the form of the

[3] We use the soft-plus function instead of the scalar exponential map for numerical stability.

[4] Since n remains constant, $\beta = \frac{1}{2}\beta_1 n$ is a constant step-size.

updates (due to Lemma 6-8 in Appx. C). For example, consider the Gaussian case in Sec. 3.1 where covariance $\mathbf{\Sigma}$ is used. Using our approach, we could easily change the parameterization to the precision $\mathbf{S} = \mathbf{\Sigma}^{-1}$ instead, by changing the parameters in (9) to

$$\begin{aligned} \boldsymbol{\tau} &:= \left\{\boldsymbol{\mu} \in \mathbb{R}^p,\ \ \mathbf{S} \in \mathcal{S}_{++}^{p\times p}\right\} \\ \boldsymbol{\lambda} &:= \left\{\boldsymbol{\mu} \in \mathbb{R}^p,\ \ \mathbf{B} \in \mathcal{R}_{++}^{p\times p}\right\} \\ \boldsymbol{\eta} &:= \left\{\boldsymbol{\delta} \in \mathbb{R}^p,\ \ \mathbf{M} \in \mathcal{S}^{p\times p}\right\}. \end{aligned} \tag{22}$$

We can use map $\mathbf{h}(\cdot)$ in the following transformations:

$$\begin{aligned} \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{S} \end{array} \right\} &= \psi(\boldsymbol{\lambda}) := \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{B}\mathbf{B}^\top \end{array} \right\} \\ \left\{ \begin{array}{c} \boldsymbol{\mu} \\ \mathbf{B} \end{array} \right\} &= \phi_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta}) := \left\{ \begin{array}{c} \boldsymbol{\mu}_t + \mathbf{B}_t^{-T}\boldsymbol{\delta} \\ \mathbf{B}_t\mathbf{h}(\mathbf{M}) \end{array} \right\}. \end{aligned} \tag{23}$$

An update (see (39) in Appx. D.1) almost identical to (16) is obtained with this parameterization and map. The difference only appears in a $O(\beta^3)$ term. Unlike the method originally described by Glasmachers et al. (2010), our formulation makes it easy for a variety of parameterizations and maps, while keeping the natural-gradient computation tractable.

To avoid computing the Hessian $\nabla_w^2 \ell(\mathbf{w})$ in Gaussian cases, we could use the re-parameterizable trick[5] for the covariance matrix (Lin et al., 2019b; 2020) in the update of (16):

$$\mathbf{g}_\Sigma = \tfrac{1}{2}\mathbb{E}_q\left[\mathbf{S}(\mathbf{w}-\boldsymbol{\mu})\nabla_w^T \ell(\mathbf{w})\right]. \tag{24}$$

By the identities in (24, 13, 15), we establish the connection of our Gaussian update to variational inference by the re-parameterizable trick, to numerical optimization by Stein's identity, and to black-box search by the REINFORCE trick.

Our approach also gives NGD updates for common univariate exponential family (EF) distributions via Auto-Differentiation (see Appx. G for the detail).

In practice, the FIM under global parameter $\boldsymbol{\tau}$ or local parameter $\boldsymbol{\eta}$ can be singular. For example, the FIM of curved EFs (Lin et al., 2019a) and MLPs (Amari et al., 2018) can be singular. The FIM of the low-rank structured Gaussian (Tran et al., 2020; Mishkin et al., 2018) has the same issue. (see Appx. J.1.6 for a discussion). We extend our approach to the following two kinds of curved EFs, where we relax Assumption 1 for local parameterizations.

In Appx. I, we adapt our local parameterization approach to a block approximation for matrix Gaussian cases, where cross-block terms in the FIM are set to zeros (see (48) in Appx. I). Our approximated FIM is guaranteed to be non-singular since matrix Gaussian is a *minimal multi-linear* EF (Lin et al., 2019a). Our approach is very different from noisy-KFAC (Zhang et al., 2018). In noisy-KFAC, KFAC approximation along with a block-approximation is used, where the approximated FIM can be singular without damping. Damping introduces an extra tuning hyper-parameter.

In Appx. H, we extend our approach to mixtures such as Gaussian mixtures using the FIM defined by the joint distribution of a mixture. For mixture distributions, Lin et al. (2019a) show that the FIM of the *joint distribution* of a *minimal conditional* mixture is guaranteed to be non-singular.

4. NGD for Structured Matrix Groups

We now show applications to NGD on matrices with special structures. The key idea is to use the fact that the auxiliary-parameter space $\mathcal{R}_{++}^{p\times p}$ used in Sec. 3 is a *general linear group* (GL group) (Belk, 2013), and structured restrictions give us its subgroups. We can specify local parameterizations for the subgroups to get a tractable NGD. We will use the Gaussian example considered in Sec. 3.5 to illustrate this idea. A similar technique could be applied to the Wishart example. We will discuss the triangular group first, and then discuss an extension inspired by the Heisenberg group.

We use $\mathcal{B}_{\text{up}}(k)$ to denote the space of following block upper-triangular p-by-p matrices as an auxiliary parameter space, where k is the block size with $0 < k < p$ and $d_0 = p - k$, and $\mathcal{D}_{++}^{d_0\times d_0}$ is the space of diagonal and invertible matrices.

$$\mathcal{B}_{\text{up}}(k) = \left\{ \begin{bmatrix} \mathbf{B}_A & \mathbf{B}_B \\ \mathbf{0} & \mathbf{B}_D \end{bmatrix} \middle| \mathbf{B}_A \in \mathcal{R}_{++}^{k\times k},\ \mathbf{B}_D \in \mathcal{D}_{++}^{d_0\times d_0} \right\}$$

When $k = 0$, $\mathcal{B}_{\text{up}}(k) = \mathcal{D}_{++}^{p\times p}$ becomes a diagonal auxiliary space. When $k = p$, $\mathcal{B}_{\text{up}}(k) = \mathcal{R}_{++}^{p\times p}$ becomes a full space. The following lemma shows $\mathcal{B}_{\text{up}}(k)$ is a *matrix group*.

Lemma 1 *$\mathcal{B}_{up}(k)$ is a matrix group that is closed under matrix multiplication.*

A local parameter space for $\mathcal{B}_{\text{up}}(k)$ is defined below with less degrees of freedom than the local space $\mathcal{S}^{p\times p}$ in (22).

$$\mathcal{M}_{\text{up}}(k) = \left\{ \begin{bmatrix} \mathbf{M}_A & \mathbf{M}_B \\ \mathbf{0} & \mathbf{M}_D \end{bmatrix} \middle| \mathbf{M}_A \in \mathcal{S}^{k\times k},\ \mathbf{M}_D \in \mathcal{D}^{d_0\times d_0} \right\}$$

where $\mathcal{D}^{d_0\times d_0}$ denotes the space of diagonal matrices. Lemma 2 shows that $\mathbf{h}(\cdot)$ defined in Sec. 3.5 is essential.

Lemma 2 *For any $\mathbf{M} \in \mathcal{M}_{up}(k)$, $\mathbf{h}(\mathbf{M}) \in \mathcal{B}_{up}(k)$.*

Using these spaces, we specify the parametrization for the Gaussian $\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \mathbf{S}^{-1})$, where the precision $\mathbf{S}$ belongs to a sub-manifold[6] of $\mathcal{S}_{++}^{p\times p}$,

$$\begin{aligned} \boldsymbol{\tau} &:= \left\{\boldsymbol{\mu} \in \mathbb{R}^p,\ \ \mathbf{S} = \mathbf{B}\mathbf{B}^T \in \mathcal{S}_{++}^{p\times p} \ \mid\ \mathbf{B} \in \mathcal{B}_{\text{up}}(k)\right\}, \\ \boldsymbol{\lambda} &:= \left\{\boldsymbol{\mu} \in \mathbb{R}^p,\ \ \mathbf{B} \in \mathcal{B}_{\text{up}}(k)\right\}, \\ \boldsymbol{\eta} &:= \left\{\boldsymbol{\delta} \in \mathbb{R}^p,\ \ \mathbf{M} \in \mathcal{M}_{\text{up}}(k)\right\}. \end{aligned}$$

[5] $\nabla_w \ell(\mathbf{w})$ is only required to exist almost surely.

[6] $\boldsymbol{\eta}$ locally gives a parametric representation of the submanifold. See (51) in Appx. J.1.3 for an equivalent global parameterization of this sub-manifold.

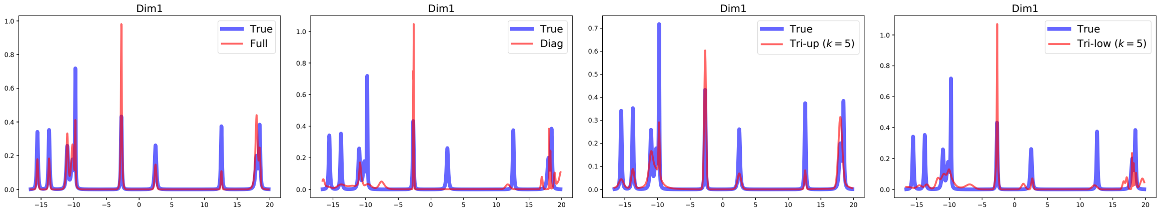


Figure 2. Comparison results of structured Gaussian mixtures to fit a 80-Dim mixture of Student's t distributions with 10 components. The first marginal dimension obtained by our updates is shown in the figure, where an upper triangular structure in the precision form achieves better approximation than a lower triangular structure and a diagonal structure. The upper triangular structure performs comparably to the full covariance structure with lower computational cost. Figure 6-8 in Appx. B show more dimensions and results on other structures.

The map $\psi \circ \phi_{\boldsymbol{\lambda}_t}(\boldsymbol{\eta})$ at $\boldsymbol{\lambda}_t := \{\boldsymbol{\mu}_t, \mathbf{B}_t\}$ is chosen to be the same as (23) due to Lemma1 and Lemma 2. Lemma 3 below shows that this local parameterization is valid.

Lemma 3 *Assumption 1-2 are satisfied in this case.*

The natural-gradients (shown in Appx. J.1.4) are

$$\hat{\boldsymbol{g}}_{\delta_0}^{(t)} = \mathbf{B}_t^{-1}\mathbf{g}_{\mu_t}; \quad \hat{\boldsymbol{g}}_{M_0}^{(t)} = \mathbf{C}_{\text{up}} \odot \kappa_{\text{up}}\big(-2\mathbf{B}_t^{-1}\mathbf{g}_{\Sigma_t}\mathbf{B}_t^{-T}\big)$$

where $\odot$ is the element-wise product, $\kappa_{\text{up}}(\mathbf{X})$ extracts non-zero entries of $\mathcal{M}_{\text{up}}(k)$ from $\mathbf{X}$ so that $\kappa_{\text{up}}(\mathbf{X}) \in \mathcal{M}_{\text{up}}(k)$, $\mathbf{J}$ is a matrix of ones, $\mathbf{C}_{\text{up}}$ is a constant matrix defined as below, where factor $\frac{1}{2}$ appears in the symmetric part of $\mathbf{C}_{\text{up}}$.

$$\mathbf{C}_{\text{up}} = \begin{bmatrix} \frac{1}{2}\mathbf{J}_A & \mathbf{J}_B \\ \mathbf{0} & \frac{1}{2}\mathbf{I}_D \end{bmatrix} \in \mathcal{M}_{\text{up}}(k)$$

The NGD update over the auxiliary parameters is

structured update

$$\boxed{\begin{aligned} \boldsymbol{\mu}_{t+1} &\leftarrow \boldsymbol{\mu}_t - \beta \mathbf{S}_t^{-1}\mathbf{g}_{\mu_t} \\ \mathbf{B}_{t+1} &\leftarrow \mathbf{B}_t \mathbf{h}\left(\beta \mathbf{C}_{\text{up}} \odot \kappa_{\text{up}}\big(2\mathbf{B}_t^{-1}\mathbf{g}_{\Sigma_t}\mathbf{B}_t^{-T}\big)\right) \quad (25) \end{aligned}}$$

where (25) preserves the structure: $\mathbf{B}_{t+1} \in \mathcal{B}_{\text{up}}(k)$ if $\mathbf{B}_k \in \mathcal{B}_{\text{up}}(k)$. When $k = p$, the update (25) recovers the update (38) of the example in Sec. 3.5 and connects to Newton's method in (16) (see (39) in Appx. D.1). When $k < p$, (25) becomes a 'structured update' preserved the group structure.

By exploiting the structure of $\mathbf{B}$ (shown in Appx. J.1.7), the update enjoys low time complexity $O(k^2p)$. The product $\mathbf{S}^{-1}\mathbf{g}_\mu$ can be computed in $O(k^2p)$. We can compute $\mathbf{Bh}(\mathbf{M})$ in $O(k^2p)$ when $\mathbf{B}$ and $\mathbf{h}(\mathbf{M})$ are block upper triangular matrices. The gradient $\mathbf{g}_\Sigma$ is obtained using Hessian where we only compute/approximate diagonal entries of the Hessian and use $O(k)$ Hessian-vector-products for non-zero entries of $\kappa_{\text{up}}\big(2\mathbf{B}^{-1}\mathbf{g}_\Sigma\mathbf{B}^{-T}\big)$ (see (53) in Appx. J.1.7). We store the non-zero entries of $\mathbf{B}$ with space complexity $O((k+1)p)$. Map $\mathbf{h}(\cdot)$ simplifies the computation and reduces the time complexity, whereas the exponential map suggested by Glasmachers et al. (2010) does not.

As shown in Appx. J.1.5, this parameterization induces a special structure over $\mathbf{S}_{\text{up}} = \mathbf{B}\mathbf{B}^T$, which is a block arrowhead matrix (O'leary & Stewart, 1990):

$$\mathbf{S}_{\text{up}} = \begin{bmatrix} \mathbf{B}_A\mathbf{B}_A^T + \mathbf{B}_B\mathbf{B}_B^T & \mathbf{B}_B\mathbf{B}_D \\ \mathbf{B}_D\mathbf{B}_B^T & \mathbf{B}_D^2 \end{bmatrix}$$

and over $\mathbf{\Sigma}_{\text{up}} = \mathbf{S}_{\text{up}}^{-1}$, which is a low-rank matrix[7]:

$$\mathbf{\Sigma}_{\text{up}} = \mathbf{U}_k\mathbf{U}_k^T + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_D^{-2} \end{bmatrix}; \quad \mathbf{U}_k = \begin{bmatrix} -\mathbf{B}_A^{-T} \\ \mathbf{B}_D^{-1}\mathbf{B}_B^T\mathbf{B}_A^{-T} \end{bmatrix}$$

where $\mathbf{U}_k$ is a rank-k matrix since $\mathbf{B}_A^{-T}$ is invertible.

As shown in Appx. J.1.8, we obtain a similar update for a block lower-triangular group $\mathcal{B}_{\text{low}}(k)$.

$$\mathcal{B}_{\text{low}}(k) = \left\{ \begin{bmatrix} \mathbf{B}_A & \mathbf{0} \\ \mathbf{B}_C & \mathbf{B}_D \end{bmatrix} \middle| \mathbf{B}_A \in \mathcal{R}_{++}^{k \times k},\ \mathbf{B}_D \in \mathcal{D}_{++}^{d_0 \times d_0} \right\}$$

Our update with a structure $\mathbf{B} \in \mathcal{B}_{\text{low}}(k)$ enjoys a low-rank structure in precision $\mathbf{S}_{\text{low}} = \mathbf{B}\mathbf{B}^T$. Likewise, our update with a structure $\mathbf{B} \in \mathcal{B}_{\text{up}}(k)$ has a low-rank structure in covariance $\mathbf{S}_{\text{up}}^{-1} = (\mathbf{B}\mathbf{B}^T)^{-1}$. These are 'structured second-order updates' where the precision can be seen as approximations of Hessians in Newton's method (see Sec. 3.2).

An extension is to construct a *hierarchical structure* inspired by the Heisenberg group (Schulz & Seesanea, 2018) by replacing a diagonal group in $\mathbf{B}_D$ with a block triangular group, where $1 < k_1 + k_2 < p$ and $d_0 = p - k_1 - k_2$

$$\mathcal{B}_{\text{up}}(k_1, k_2) = \left\{ \begin{bmatrix} \mathbf{B}_A & \mathbf{B}_B \\ \mathbf{0} & \mathbf{B}_D \end{bmatrix} \middle| \mathbf{B}_D = \begin{bmatrix} \mathbf{B}_{D_1} & \mathbf{B}_{D_2} \\ \mathbf{0} & \mathbf{B}_{D_4} \end{bmatrix} \right\}$$

where $\mathbf{B}_A \in \mathcal{R}_{++}^{k_1 \times k_1}$, $\mathbf{B}_{D_1} \in \mathcal{D}_{++}^{d_0 \times d_0}$, $\mathbf{B}_{D_4} \in \mathcal{R}_{++}^{k_2 \times k_2}$.

This group has a flexible structure and recovers the block triangular group as a special case when $k_2 = 0$. Likewise, We can define a lower block Heisenberg group $\mathcal{B}_{\text{low}}(k_1, k_2)$. In Appx. J.2, we show that these groups can be used as structured parameter spaces, which could be useful for problems of interest in optimization, inference, and search.

If the Hessian $\nabla^2\ell(\mathbf{w})$ has a model-specific structure, we could design a customized group to capture such structure in the precision. For example, the Hessian of layer-wise matrix weights of a neural network admits a Kronecker form

[7] The zero block highlighted in red in the expression of $\mathbf{\Sigma}_{\text{up}}$ guarantees the FIM to be non-singular (see Appx. J.1.6).

(see Appx. I.2). We can use a *Kronecker product group* to capture such structure. The Kronecker structure is preserved even when the Gauss-Newton approximation is employed. This group structure can further reduce the time complexity from the quadratic complexity to a linear complexity in k (see Appx. I.1 and Figure 4).

In general, many subgroups (e.g., block (invertible) triangular Toeplitz groups and groups constructed from an existing group via the group conjugation by an element of the rotation group) of the GL group $\mathbb{R}_{++}^{p\times p}$ can be used as structured auxiliary parameter spaces $\mathcal{B}$. Our approach to construct a structured Gaussian-precision is valid if there exists a local parameter space $\mathcal{M}$ so that $\mathbf{h}(\mathbf{M}) \in \mathcal{B}$ for any $\mathbf{M} \in \mathcal{M}$ and Assumptions 1-3 are satisfied. If these conditions hold, the inverse of FIM $\mathbf{F}_{\boldsymbol{\eta}}^{-1}(\boldsymbol{\eta}_0)$ using $\mathcal{M}$ will be easy to compute due to Lemma 11 in Appx. D.1. We can even weaken Assumption 1 as discussed in Sec. 3.5. The computational requirements are (1) group product and inverse can be efficiently implemented and (2) $\kappa\big(2\mathbf{B}^{-1}\mathbf{g}_\Sigma\mathbf{B}^{-T}\big) \in \mathcal{M}$ can be implemented without the whole Hessian in (15), where $\kappa(\cdot)$ converts $\mathbb{R}^{p\times p}$ to $\mathcal{M}$.

5. Numerical Results

We present results on problems involving search, inference, optimization, and deep learning, where Table 1 in Appx. A summarizes our updates. We use $\mathbf{h}(\cdot)$ defined in Sec. 3.5 to replace the matrix exponential map in our proposed updates.

5.1. Search with Re-parameterizable Gradients

We validate our update in the metric nearness task (Brickell et al., 2008) using a Wishart distribution as a search distribution q with $\gamma = 0$ in (1). The objective function is $\ell(\mathbf{W}) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathbf{W}\mathbf{Q}\mathbf{x}_i - \mathbf{x}_i\|_2^2$, where $\mathbf{x}_i \in \mathcal{R}^d$, $\mathbf{Q} \in \mathcal{S}_{++}^{d\times d}$ and $\mathbf{W} \in \mathcal{S}_{++}^{d\times d}$. The optimal solution is $\mathbf{Q}^{-1}$. We randomly generate $\mathbf{x}_i$ and $\mathbf{Q}$ with $d = 50$, $N_{\text{train}} = 125,000$ for training and $N_{\text{test}} = 25,000$ for testing. All methods are trained using mini-batches, where the size of mini-batch is 100. We use re-parameterizable gradients with 1 Monte Carlo (MC) sample in our update (referred to as "our-rep"), where we update $\mathbf{B}$ and b. we also consider to only update $\mathbf{B}$ with re-parameterizable gradients (referred to as "our-fixed-rep"). To numerically show the similarity between RGD and our update, we consider a case where gradients are evaluated at the mean (referred to as "-mean"). We consider baseline methods: the RGD update for positive-definite manifolds and the Riemannian trivialization[8] (Lezcano Casado, 2019), where gradients are evaluated at the mean. For the trivialization method, we consider trivializations for the positive-definite manifold: a Cholesky factor and the matrix logarithmic function. We report the best result of the trivializations denoted by "Adam", where we use Adam to perform updates in a trivialized (Euclidean) space. From Figure 3a, we can see our update performs similarly to RGD if gradients are evaluated at the mean while the trivialization method is trapped in a local mode. If we use re-parameterizable gradients, jointly updating both parameters is better than only updating $\mathbf{B}$.

5.2. Variational Inference with Gaussian Mixtures

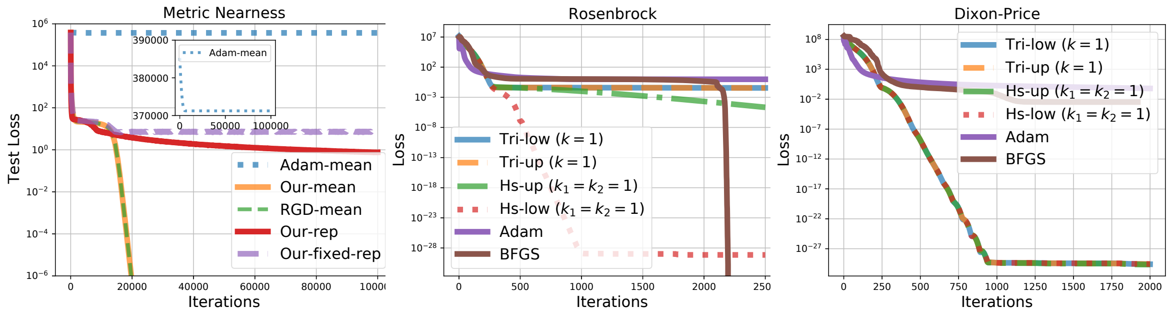
We consider the Gaussian mixture approximation problem (Lin et al., 2020), where we use a Gaussian mixture with K components $q(\mathbf{w}) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})$ as a variational distribution q with $\gamma = 1$ in (1). The goal of the problem is to approximate a mixture of d-dimensional Student's t distributions $\exp(-\ell(\mathbf{w})) = \frac{1}{C}\sum_{c=1}^{C}\mathcal{T}(\mathbf{w}|\mathbf{u}_c, \mathbf{V}_c, \alpha)$ with $\alpha = 2$. We consider six kinds of structures of each Gaussian component: full precision (referred to as "full"), diagonal precision (referred to as "diag"), precision with the block upper triangular structure (referred to as "Tri-up"), precision with the block lower triangular structure (referred to as "Tri-low"), precision with the block upper Heisenberg structure (referred to as "Hs-up"), precision with the block lower Heisenberg structure (referred to as "Hs-low"). Each entry of $\mathbf{u}_c$ is generated uniformly in an interval $(-s, s)$. Each matrix $\mathbf{V}_c$ is generated as suggested by Lin et al. (2020). We consider a case with $K = 40, C = 10, d = 80, s = 20$. We update each component during training, where 10 MC samples are used to compute gradients. We compute gradients as suggested by Lin et al. (2020), where second-order information is used. For structured updates, we compute Hessian-vector products and diagonal entries of the Hessian without directly computing the Hessian $\nabla_w^2\ell(\mathbf{w})$. From Figure 2, we can see an *upper structure* is better for inference problems[9]. Figure 6-8 in Appx. B show more results on dimensions and structures such as Heisenberg structures.

5.3. Structured Second-order Optimization

We consider non-separable valley-shaped test functions for optimization: Rosenbrock: $\ell_{\text{rb}}(\mathbf{w}) = \frac{1}{d}\sum_{i=1}^{d-1}\big[100(w_{i+1} - w_i)^2 + (w_i - 1)^2\big]$, and Dixon-Price: $\ell_{\text{dp}}(\mathbf{w}) = \frac{1}{d}\big[(w_i - 1)^2 + \sum_{i=2}^{d} i(2w_i^2 - w_{i-1})^2\big]$. We test our structured Newton's updates, where we set $d = 200$ and $\gamma = 1$ in (1). We consider these structures in the precision: the upper triangular structure (denoted by "Tri-up"), the lower triangular structure (denoted by "Tri-low"), the upper Heisenberg structure (denoted by "Hs-up"), and the lower Heisenberg structure (denoted by "Hs-low"), where second-order information is used. For our updates, we compute Hessian-vector products

[8] In variational inference (VI), trivializing a parametric distribution is known as black-box VI (Ranganath et al., 2014).

[9] For variational inference, an *upper structure* in the precision is better than a lower structure to capture off-diagonal correlations.



(a) (b) (c)

Figure 3. The performances of our updates for search and optimization problems. Figure 3a shows the performances using a Wishart distribution to search the optimal solution of a metric nearness task where our method evaluated at the mean behaves like RGD and converges faster than the Riemannian trivialization (Lezcano Casado, 2019) with Adam. Our updates with re-parameterizable gradients also can find a solution near the optimal solution. Figure 3b and 3c show the performances using structured Newton's updates to optimize non-separable, valley-shaped, 200-dimensional functions, where our updates only require to compute diagonal entries of Hessian and Hessian-vector products. Our updates with a lower Heisenberg structure in the precision form converge faster than BFGS and Adam.

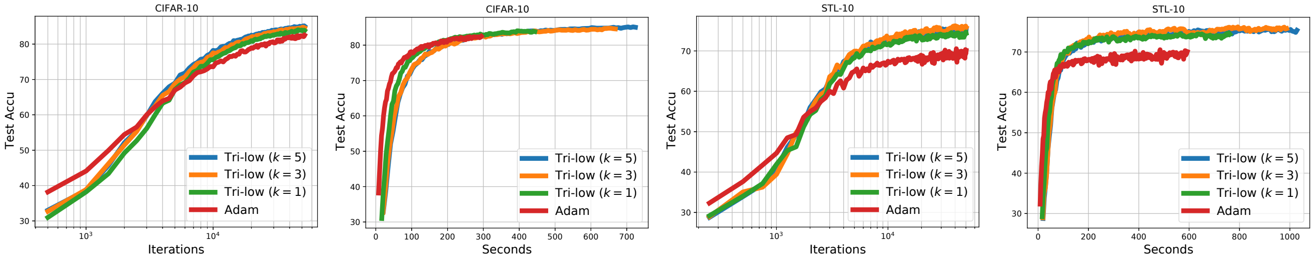


Figure 4. The performances for optimization of a CNN using matrix Gaussian with low-rank in a Kronecker precision form, where our updates ($O(k|\mathbf{w}|)$) have a linear iteration cost like Adam ($O(|\mathbf{w}|)$) and are automatically parallelized by Auto-Diff. Our updates achieve higher test accuracy (75.8% on "STL-10" and 85.0% on "CIFAR-10") than Adam (69.5% on "STL-10" and 82.3% on "CIFAR-10").

and diagonal entries of the Hessian without directly computing the Hessian. We consider baseline methods: the BFGS method provided by SciPy and the Adam optimizer, where the step-size is tuned for Adam. We evaluate gradients at the mean for all methods. Figure 3b-3c show the performances of all methods[10], where our updates with a lower Heisenberg structure converge faster than BFGS and Adam.

5.4. Optimization for Deep Learning

We consider a CNN model with 9 hidden layers, where 6 layers are convolution layers. For a smooth objective, we use average pooling and GELU (Hendrycks & Gimpel, 2016) as activation functions. We employ L_2 regularization with weight 10^{-2}. We set $\gamma = 1$ in (1) in our updates. We train the model with our updates derived from matrix Gaussian (see Appx. I) for each layer-wise matrix weight[11] on datasets "CIFAR-10", "STL-10". Each Gaussian-precision has a Kronecker product group structure of two lower-triangular groups (referred to as "Tri-low") for computational complexity reduction (see Appx. I.1). For "CIFAR-10" and "STL-10", we train the model with mini-batch size 20. Additional results on "CIFAR-100" can be found at Figure 5 in Appx. B. We evaluate gradients at the mean and approximate the Hessian by the Gauss-Newton approximation. We compare our updates to Adam, where the step-size for each method is tuned by grid search. We use the same initialization and hyper-parameters in all methods. We report results in terms of test accuracy, where we average the results over 5 runs with distinct random seeds. From Figure 4, we can see our structured updates have a linear iteration cost like Adam while achieve higher test accuracy.

6. Conclusion

We propose a tractable NGD for structured spaces. The method enables more flexible covariance structures with lower complexity than other methods. Preliminarily results show the method is promising. An interesting direction is to evaluate its performance on large-scale problems.

Acknowledgements

WL is supported by a UBC International Doctoral Fellowship. This research was partially supported by the Canada CIFAR AI Chair Program.

[10] Empirically, we find out that a *lower structure* in the precision performs better than an upper structure for optimization tasks including optimization for neural networks.

[11] $\mathbf{W} \in \mathbb{R}^{c_{\text{out}} \times c_{\text{in}} p^2}$ is a weight matrix, where p, c_{in}, c_{out} are the kernel size, the number of input, output channels, respectively.

References

Agakov, F. V. and Barber, D. An auxiliary variational method. In *International Conference on Neural Information Processing*, pp. 561–566. Springer, 2004.

Akimoto, Y. and Hansen, N. Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pp. 197–204, 2016.

Amari, S.-i., Ozeki, T., Karakida, R., Yoshida, Y., and Okada, M. Dynamics of learning in mlp: Natural gradient and singularity revisited. *Neural computation*, 30(1): 1–33, 2018.

Baba, N. Convergence of a random optimization method for constrained optimization problems. *Journal of Optimization Theory and Applications*, 33(4):451–461, 1981.

Belk, J. Lecture Notes: Matrix Groups. http://faculty.bard.edu/belk/math332/MatrixGroups.pdf, 2013. Accessed: 2021/02.

Beyer, H.-G. *The theory of evolution strategies*. Springer Science & Business Media, 2001.

Brickell, J., Dhillon, I. S., Sra, S., and Tropp, J. A. The metric nearness problem. *SIAM Journal on Matrix Analysis and Applications*, 30(1):375–396, 2008.

Chen, S.-W., Chou, C.-N., and Chang, E. Y. Ea-cg: An approximate second-order method for training fully-connected neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3337–3346, 2019.

Dangel, F., Harmeling, S., and Hennig, P. Modular block-diagonal curvature approximations for feedforward architectures. In *International Conference on Artificial Intelligence and Statistics*, pp. 799–808. PMLR, 2020.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pp. 441–452, 2018.

Glasmachers, T., Schaul, T., Yi, S., Wierstra, D., and Schmidhuber, J. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 393–400, 2010.

Graves, A. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

Hazan, E., Levy, K. Y., and Shalev-Shwartz, S. On graduated optimization for stochastic non-convex problems. In *International conference on machine learning*, pp. 1833–1841. PMLR, 2016.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Hosseini, R. and Sra, S. Matrix manifold optimization for Gaussian mixtures. In *Advances in Neural Information Processing Systems*, pp. 910–918, 2015.

Johansen, S. Introduction to the theory of regular exponential famelies. 1979.

Khan, M. and Lin, W. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Artificial Intelligence and Statistics*, pp. 878–887, 2017.

Khan, M. E. and Nielsen, D. Fast yet Simple Natural-Gradient Descent for Variational Inference in Complex Models. *arXiv preprint arXiv:1807.04489*, 2018.

Khan, M. E. and Rue, H. Learning-algorithms from Bayesian principles. 2020. https://emtiyaz.github.io/papers/learning_from_bayes.pdf.

Khan, M. E., Lin, W., Tangkaratt, V., Liu, Z., and Nielsen, D. Variational adaptive-Newton method for explorative learning. *arXiv preprint arXiv:1711.05560*, 2017.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2611–2620, 2018.

Leordeanu, M. and Hebert, M. Smoothing-based optimization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2008.

Lezcano Casado, M. Trivializations for gradient-based optimization on manifolds. *Advances in Neural Information Processing Systems*, 32:9157–9168, 2019.

Li, Z. and Zhang, Q. A simple yet efficient evolution strategy for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation*, 22(5):637–646, 2017.

Lin, W. An upper triangular version of the cholesky decompostion. https://math.stackexchange.com/q/4114067, 2021. Accessed: 2021/04.

Lin, W., Khan, M. E., and Schmidt, M. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*, pp. 3992–4002, 2019a.

Lin, W., Khan, M. E., and Schmidt, M. Stein's Lemma for the Reparameterization Trick with Exponential-family Mixtures. *arXiv preprint arXiv:1910.13398*, 2019b.

Lin, W., Schmidt, M., and Khan, M. E. Handling the positive-definite constraint in the bayesian learning rule. In *International Conference on Machine Learning*, pp. 6116–6126. PMLR, 2020.

Malagò, L., Matteucci, M., and Pistone, G. Towards the geometry of estimation of distribution algorithms based on the exponential family. In *Proceedings of the 11th workshop proceedings on Foundations of genetic algorithms*, pp. 230–242, 2011.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M., and Khan, M. E. SLANG: Fast Structured Covariance Approximations for Bayesian Deep Learning with Natural Gradient. In *Advances in Neural Information Processing Systems*, pp. 6246–6256, 2018.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Mobahi, H. and Fisher III, J. A theoretical analysis of optimization by gaussian continuation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

O'leary, D. and Stewart, G. Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. *Journal of Computational Physics*, 90(2):497–505, 1990.

Opper, M. and Archambeau, C. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

Osawa, K., Swaroop, S., Khan, M. E. E., Jain, A., Eschenhagen, R., Turner, R. E., and Yokota, R. Practical deep learning with Bayesian principles. In *Advances in neural information processing systems*, pp. 4287–4299, 2019a.

Osawa, K., Tsuji, Y., Ueno, Y., Naruse, A., Yokota, R., and Matsuoka, S. Large-scale distributed second-order optimization using kronecker-factored approximate curvature for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12359–12367, 2019b.

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

Ros, R. and Hansen, N. A simple modification in cma-es achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*, pp. 296–305. Springer, 2008.

Salimbeni, H., Eleftheriadis, S., and Hensman, J. Natural Gradients in Practice: Non-Conjugate Variational Inference in Gaussian Process Models. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Schulz, E. and Seesanea, A. Extensions of the Heisenberg group by two-parameter groups of dilations. *arXiv preprint arXiv:1804.10305*, 2018.

Smith, W., Hocking, R., et al. Wishart variate generator. *Applied Statistics*, 21:341–345, 1972.

Spall, J. C. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

Sun, K. and Nielsen, F. Relative fisher information and natural gradient for learning large modular models. In *International Conference on Machine Learning*, pp. 3289–3298, 2017.

Sun, Y., Wierstra, D., Schaul, T., and Schmidhuber, J. Efficient natural evolution strategies. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 539–546, 2009.

Sun, Y., Schaul, T., Gomez, F., and Schmidhuber, J. A linear time natural evolution strategy for non-separable functions. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pp. 61–62, 2013.

Sutton, R. S., Barto, A. G., et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Teboulle, M. Entropic proximal mappings with applications to nonlinear programming. *Math. Oper. Res.*, 17(3): 670âĂŞ690, August 1992. ISSN 0364-765X.

Tran, M.-N., Nguyen, N., Nott, D., and Kohn, R. Bayesian deep net glm and glmm. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. Natural evolution strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3381–3387. IEEE, 2008.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Williams, R. J. and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Zellner, A. Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association*, 81(394):446–451, 1986.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pp. 5847–5856, 2018.