Practical Structured Riemannian Optimization with Momentum by using Generalized Normal Coordinates

Abstract

Adding momentum into Riemannian optimization is computationally challenging due to the intractable ODEs needed to define the exponential and parallel transport maps. We address these issues by proposing new local coordinates to exploit sparse structures and efficiently approximate the ODEs, which results in a numerically stable update scheme. Our approach extends the structured NGD method of Lin et al. (2021a) by incorporating momentum into it and scaling the method for large-scale applications arising in numerical optimization, deep learning, variational inference, and evolution strategies.

Keywords: Riemannian Manifolds, Matrix Lie Groups, Numerical Optimization

1. Introduction and Background

Riemannian gradient methods are useful for optimization on manifolds. Momentum is a common approach to improve gradient-based methods. Unfortunately, adding momentum into Riemannian optimization is computationally intensive due to the intractable ODEs. Moreover, Riemannian methods could perform poorly and be numerically unstable in single-precision floating-point settings such as training large-scale neural networks. Straightforward approximations of the ODEs in global coordinates involves computing nonsparse and high-order derivatives which limits the large-scale usage of Riemannian methods.

Extended Abstract Track

In this work, we propose special local coordinates to simplify the approximations and exploit sparse structures, which leads to numerically stable update schemes. We apply our approach to extend the structured NGD method of Lin et al. (2021a) by introducing momentum to the method, and scaling the structured NGD for large-scale applications. We will begin with the basics, introduce our proposed method, and conclude with an example on Gaussian manifolds. More examples will be discussed in the appendices.

1.1. Riemannian Gradient Descent and Natural-gradient Descent

Consider a geodesically complete manifold $\mathcal{M}$ with the Fisher-Rao $\mathbf{F}$ metric. The metric is known as the Fisher information matrix and defined in A.1. Under a global parametrization $\boldsymbol{\tau}$, Riemannian gradient descent (Bonnabel, 2013) (RGD) is defined as

$$\boldsymbol{\tau} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}, -\alpha \mathbf{F}_\tau^{-1}(\boldsymbol{\tau})\mathbf{g}(\boldsymbol{\tau})), \tag{1}$$

where $\mathbf{v} := \mathbf{F}_\tau^{-1}(\boldsymbol{\tau})\mathbf{g}(\boldsymbol{\tau})$ is a Riemannian gradient, $\mathbf{g}(\boldsymbol{\tau})$ is a Euclidean gradient (A.K.A., 1-form), $\mathbf{F}_\tau$ is the metric, $\mathrm{Exp}(\boldsymbol{\tau}, \mathbf{v})$ is the Riemannian exponential map (defined in A.3) induced by the geodesic ODE at current $\boldsymbol{\tau}$ with Riemannian direction $\mathbf{v}$, and α is a step-size. The exponential map often does not admit a closed-form expression. For an approximation of the map, we have to evaluate the Christoffel symbols defined at A.2, which involves computing partial derivatives of the Fisher matrix.

Natural-gradient descent (NGD) is a special case of RGD in (1) that uses a first-order Taylor approximation of the exponential map to avoid evaluating the Christoffel symbols,

$$\boldsymbol{\tau} \leftarrow= \boldsymbol{\tau} - \alpha \mathbf{F}_\tau^{-1}(\boldsymbol{\tau})\mathbf{g}(\boldsymbol{\tau}) = \mathrm{Ret}(\boldsymbol{\tau}, -\alpha \mathbf{F}_\tau^{-1}(\boldsymbol{\tau})\mathbf{g}(\boldsymbol{\tau})), \tag{2}$$

© 2022 .

This approximation, which is known as the Euclidean retraction, is defined as,

$$\mathrm{Exp}(\mathbf{x}, \boldsymbol{\nu}) = \mathbf{r}(1) \approx \mathrm{Ret}(\mathbf{x}, \boldsymbol{\nu}) := \mathbf{r}(0) + \dot{\mathbf{r}}(0)(1-0) = \mathbf{x} + \boldsymbol{\nu},$$

where $\mathbf{r}(t)$ is the solution of the geodesic ODE with initial values $\mathbf{r}(0) = \mathbf{x}$ and $\dot{\mathbf{r}}(0) = \boldsymbol{\nu}$.

1.2. Standard Riemannian Normal Parametrization

We now discuss updates using a special local parametrization that will allow us to reduce computational costs, and we will generalize this later.

Let $\boldsymbol{\tau}$ be the global parametrization, and $\boldsymbol{\eta}$ be a Riemannian normal (local) parametrization at the current point $\boldsymbol{\tau}^{(\mathrm{cur})}$ so that $\boldsymbol{\tau} = \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, \mathbf{F}_\tau^{-1/2}(\boldsymbol{\tau}^{(\mathrm{cur})})\boldsymbol{\eta})$, where $\boldsymbol{\tau}^{(\mathrm{cur})}$ and $\mathbf{F}_\tau^{-1/2}(\boldsymbol{\tau}^{(\mathrm{cur})})$ are constants. In the $\boldsymbol{\eta}$ coordinates, the origin $\boldsymbol{\eta}_0 := \mathbf{0}$ represents point $\boldsymbol{\tau}^{(\mathrm{cur})}$.

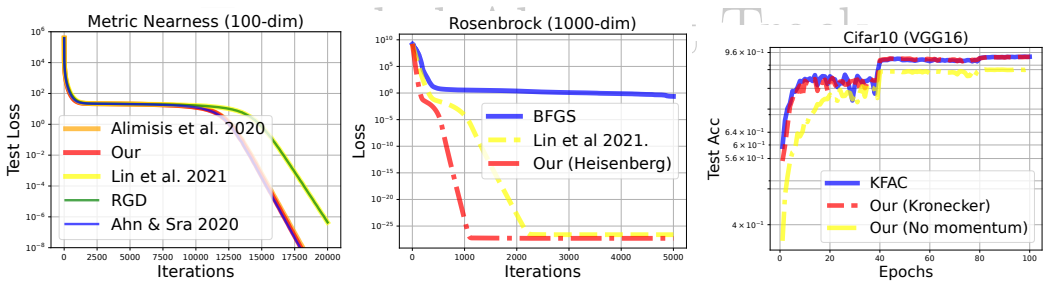
The following properties hold at point $\boldsymbol{\tau}^{(\mathrm{cur})}$ in local coordinate $\boldsymbol{\eta}$ at $\boldsymbol{\eta}_0 = \mathbf{0}$.

$$\boldsymbol{\tau}^{(\mathrm{cur})} = \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, \mathbf{F}_\tau^{-1/2}(\boldsymbol{\tau}^{(\mathrm{cur})})\boldsymbol{\eta}_0); \;\; \mathbf{F}_\eta(\boldsymbol{\eta}_0) = \mathbf{I}; \;\; \Gamma^a_{cb}(\boldsymbol{\eta}_0) = 0. \tag{3}$$

The update of RGD in Eq. 1 can be written as a NGD update in local coordinate $\boldsymbol{\eta}$:

$$\left.\begin{aligned} &\boldsymbol{\eta}_1 \leftarrow \boldsymbol{\eta}_0 - \alpha \mathbf{g}(\boldsymbol{\eta}_0), \\ &\boldsymbol{\tau}^{(\mathrm{new})} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, \mathbf{F}_\tau^{-1/2}(\boldsymbol{\tau}^{(\mathrm{cur})})\boldsymbol{\eta}_1), \end{aligned}\right\} \equiv \boldsymbol{\tau} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}, -\alpha \mathbf{F}_\tau^{-1}(\boldsymbol{\tau})\mathbf{g}(\boldsymbol{\tau}), \tag{4}$$

where $\boldsymbol{\eta}_0 = \mathbf{0}$, $\mathbf{F}_\eta(\boldsymbol{\eta}_0) = \mathbf{I}$ and $\mathbf{g}(\boldsymbol{\eta}_0) = \mathbf{F}_\tau^{-1/2}(\boldsymbol{\tau}^{(\mathrm{cur})})\mathbf{g}(\boldsymbol{\tau}^{(\mathrm{cur})})$ is a Euclidean gradient. This update remains on the manifold while NGD in global coordinate $\boldsymbol{\tau}$ (see Eq. 2) does not especially when $\boldsymbol{\tau}$ is in a constraint set.



(a) Manifold Opt. (C.1) (b) Numerical Opt. (C.2) (c) Deep Learning (C.3)

Figure 1: Preliminary Numerical Results (See App. C for the description)

2. Natural Gradient Descent with Momentum

2.1. Momentum in Riemannian Gradient Descent

To incorporate momentum in RGD, we have to use a parallel transport map to move a (gradient) vector at the current point $\boldsymbol{\tau}^{(\mathrm{cur})}$ to a new point $\boldsymbol{\tau}^{(\mathrm{new})}$ on a curved space. This map is induced by the transport ODE. We use the transport map $\Gamma_{\tau^{(\mathrm{cur})} \to \tau^{(\mathrm{new})}}$ (defined in A.5) to transport a Euclidean gradient at $\tau^{(\mathrm{cur})}$ to a Euclidean gradient at $\tau^{(\mathrm{new})}$. We consider the following update with momentum weight β under global coordinate $\boldsymbol{\tau}$.

$$\begin{aligned} &\mathbf{m}^{(\mathrm{cur})} \leftarrow \beta \mathbf{w}^{(\mathrm{cur})} + \sqrt{\alpha}\mathbf{g}(\boldsymbol{\tau}^{(\mathrm{cur})}), \\ &\boldsymbol{\tau}^{(\mathrm{new})} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, -\sqrt{\alpha}\mathbf{F}_\tau^{-1}(\boldsymbol{\tau}^{(\mathrm{cur})})\mathbf{m}^{(\mathrm{cur})}), \\ &\mathbf{w}^{(\mathrm{new})} \leftarrow \Gamma_{\tau^{(\mathrm{cur})} \to \tau^{(\mathrm{new})}}(\mathbf{m}^{(\mathrm{cur})}), \end{aligned} \tag{5}$$

where $\mathbf{w}$ is a Euclidean gradient and $\mathbf{m}$ is a momentum for Euclidean gradients.

This update is equivalent to the update of Alimisis et al. (2020) described in A.7. In Alimisis et al. (2020), the authors use the parallel transport map for Riemannian gradients instead (see A.4). However, these two updates become distinct when we approximate any of the transport maps. In this work, we choose to approximate the parallel transport map for Euclidean gradients as it is computationally challenging to compute exactly.

2.2. Generalized Riemannian Normal Parametrizations

Our idea is to use a generalized Riemannian normal parametrization to simplify the computation. The standard Riemannian normal parametrization is computationally challenging since it involves with the computation of the Riemannian exponential map.

To get rid of the dependence on the map, we define a generalized Riemannian normal coordinate $\boldsymbol{\eta}$ associated to point $\boldsymbol{\tau}^{(\mathrm{cur})}$ via a diffeomorphic map $\boldsymbol{\tau} = \boldsymbol{\phi}_{\tau^{(\mathrm{cur})}}(\boldsymbol{\eta})$ such that the following conditions hold, where the Christoffel symbols need not vanish at $\boldsymbol{\eta}_0 = \mathbf{0}$.

$$\boldsymbol{\tau}^{(\mathrm{cur})} = \boldsymbol{\phi}_{\tau^{(\mathrm{cur})}}(\boldsymbol{\eta}_0); \;\; \mathbf{F}_\eta(\boldsymbol{\eta}_0) = \mathbf{I}. \tag{6}$$

We recover the standard normal coordinate if the exponential map is used as the diffeomorphic map. However, we can instead choose a more computationally efficient diffeomorphic map. Moreover, we often can efficiently compute the Christoffel symbols at $\boldsymbol{\eta}_0$ if necessary.

2.3. NGD with Momentum using Generalized Normal Coordinates

Extended Abstract Track

We propose a practical approximation of Eq. 5 by using local coordinate $\boldsymbol{\eta}$ associated to $\boldsymbol{\tau}^{(\mathrm{cur})}$, where $\boldsymbol{\tau}$ is a global coordinate.

$$\begin{aligned}
&\mathbf{m}^{(\eta_0)} \leftarrow \beta \mathbf{w}^{(\eta_0)} + \sqrt{\alpha}\mathbf{g}(\boldsymbol{\eta}_0), \\
&\left.\begin{aligned}
&\boldsymbol{\eta}_1 && \leftarrow \boldsymbol{\eta}_0 - \sqrt{\alpha}\mathbf{F}_\eta^{-1}(\eta_0)\mathbf{m}^{(\eta_0)}, \\
&\boldsymbol{\tau}^{(\mathrm{new})} && \leftarrow \boldsymbol{\phi}_{\tau^{(\mathrm{cur})}}(\boldsymbol{\eta}_1),
\end{aligned}\right\} \overbrace{\approx}^{\text{see Eq. 4}} \boldsymbol{\tau}^{(\mathrm{new})} \leftarrow \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, -\sqrt{\alpha}\mathbf{F}_\tau^{-1}(\boldsymbol{\tau}^{(\mathrm{cur})})\mathbf{m}^{(\mathrm{cur})}), \\
&\mathbf{w}^{(\eta_1)} \leftarrow \Gamma_{\eta_0 \to \eta_1}(\mathbf{m}^{(\eta_0)}),
\end{aligned}$$

where $\mathbf{F}_\eta^{-1}(\eta_0) = \mathbf{I}$ and $\boldsymbol{\eta}_0 = \mathbf{0}$. Note that $\boldsymbol{\eta}_0$ represents $\boldsymbol{\tau}^{(\mathrm{cur})}$ while $\boldsymbol{\eta}_1$ represents $\boldsymbol{\tau}^{(\mathrm{new})}$.

To use normal coordinate $\boldsymbol{\xi}$ associated to $\boldsymbol{\tau}^{(\mathrm{new})}$ at the next iteration, we have to transform the Euclidean gradient $\mathbf{w}^{(\eta_1)}$ in coordinate η into coordinate $\boldsymbol{\xi}$ via the chain rule:

$$\mathbf{w}^{(\xi_0)} = \mathbf{J}(\boldsymbol{\xi}_0)\mathbf{w}^{(\eta_1)}; \;\; \mathbf{J}(\boldsymbol{\xi}) := \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\xi}},$$

where $\boldsymbol{\xi}_0 = \mathbf{0}$, $\mathbf{J}(\boldsymbol{\xi})$ is the Jacobian, and both $\boldsymbol{\xi}_0$ and $\boldsymbol{\eta}_1$ represent $\boldsymbol{\tau}^{(\mathrm{new})}$.

We can compute the Jacobian as $\mathbf{J}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{\phi}^{-1}_{\tau^{(\mathrm{cur})}} \circ \boldsymbol{\phi}_{\tau^{(\mathrm{new})}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ since by construction we have $\boldsymbol{\tau} = \boldsymbol{\phi}_{\tau^{(\mathrm{cur})}}(\boldsymbol{\eta}) = \boldsymbol{\phi}_{\tau^{(\mathrm{new})}}(\boldsymbol{\xi})$. In practice, we could also directly compute $\mathbf{w}^{(\xi_0)}$ by automatic-differentiation. As mentioned in Sec. 2.1, we use the transport map for Euclidean gradients so that the Jacobian can be easy to computed at $\boldsymbol{\xi}_0 = \mathbf{0}$. If the transport map for Riemannian gradients is used, we have to compute the inverse of the Jacobian.

We now discuss approximations of the transport map for Euclidean gradients.

2.4. Approximations of the Transport Map

To avoid evaluating the Christoffel symbols, we could use a zero-order Taylor expansion to approximate the parallel transport map as shown below,

$$\Gamma_{\eta_0 \to \eta_1}(\mathbf{m}^{(\eta_0)}) = \boldsymbol{\omega}(1) \approx \boldsymbol{\omega}(0) = \mathbf{m}^{(\eta_0)}, \tag{7}$$

where $\boldsymbol{\omega}(t)$ is the solution of the transport ODE with initial value $\boldsymbol{\omega}(0) = \mathbf{m}^{(\eta_0)}$.

A more accurate approximation is a first-order approximation as in A.6, where we have to compute the Christoffel symbols at $\boldsymbol{\eta}_0 = \mathbf{0}$. We often can simplify the computation since $\boldsymbol{\eta}$ is a generalized Riemannian normal coordinate. We give an example in the next section.

3. Gaussian Fisher-Rao Manifolds

We will extend the local coordinates in the method of Lin et al. (2021a) by using the generalized normal coordinates. Our extension allows the method to incorporate momentum and improves its performance for large-scale applications. Due to the space constraint, we only consider full Gaussians with constant mean. Other cases are given in Appendix B.

Consider a d-dimension Gaussian family with zero mean $q(\mathbf{z}|\boldsymbol{\tau})$, where the global coordinate $\boldsymbol{\tau}$ is the inverse covariance, which is symmetric positive-definite. For efficiency, we propose a new re-parameterization instead of the one in Lin et al. (2021a).

We introduce an auxiliary parameterization $\mathbf{B}$ such that $\boldsymbol{\tau} = \mathbf{B}^T\mathbf{B}$, where $\mathbf{B} \in \mathrm{GL}^{d\times d}$ is an invertible matrix. The auxiliary parameter $\mathbf{B}$ has an additional Lie group structure.

We define a local parameterization $\boldsymbol{\eta}$ associated to $\mathbf{B}^{(\mathrm{cur})}$ as $\mathbf{B} = \mathbf{h}(\frac{1}{\sqrt{2}}\boldsymbol{\eta})\mathbf{B}^{(\mathrm{cur})}$, where $\mathbf{h}(\mathbf{N})$ is the matrix exponential (A.K.A., the Lie-group exponential map) $\mathrm{Expm}(\mathbf{N}) := \sum_{n=0}^{\infty} \frac{N^n}{n!}$ or its truncation, and $\boldsymbol{\eta}$ is a symmetric matrix. Indeed, local parameter $\boldsymbol{\eta}$ lives in a subspace of the Lie algebra of $\mathbf{B}$ with the matrix commutator as its Lie bracket.

Extended Abstract Track

Note that $\boldsymbol{\eta}$ is a generalized Riemannian normal coordinate associated to $\boldsymbol{\tau}^{(\mathrm{cur})}$ via diffeomorphic map $\boldsymbol{\tau} = (\mathbf{B}^{(\mathrm{cur})})^T\mathbf{h}^T(\frac{1}{\sqrt{2}}\boldsymbol{\eta})\mathbf{h}(\frac{1}{\sqrt{2}}\boldsymbol{\eta})\mathbf{B}^{(\mathrm{cur})}$, where $\boldsymbol{\tau}^{(\mathrm{cur})} = (\mathbf{B}^{(\mathrm{cur})})^T\mathbf{B}^{(\mathrm{cur})}$. Moreover, $\mathbf{F}_\eta(\boldsymbol{\eta}_0) = \mathbf{I}$, where $\boldsymbol{\eta}_0 = \mathbf{0}$. In this case, the Christoffel symbols at $\boldsymbol{\eta}_0$ vanish. We write $\mathbf{w}$ as $\mathbf{w} = \mathbf{w}^{(\xi_0)} = \mathbf{w}^{(\eta_1)}$ since the Jacobian matrix evaluated at $\boldsymbol{\xi}_0$ is the identity.

The proposed update is simplified as below, where we can efficiently compute the Euclidean gradient $\mathbf{g}(\boldsymbol{\eta}_0)$ and use the zero-order approximation for the transport. Since the Christoffel symbols vanish, this approximation is also the first-order approximation.

$$\begin{aligned}
&\mathbf{m} \leftarrow \beta\mathbf{w} + \sqrt{\alpha}\mathbf{g}(\boldsymbol{\eta}_0),\\
&\boldsymbol{\eta}_1 \leftarrow \boldsymbol{\eta}_0 - \sqrt{\alpha}\mathbf{m},\\
&\left.\begin{aligned}
&\mathbf{B}^{(\mathrm{new})} \leftarrow \mathbf{h}(\tfrac{1}{\sqrt{2}}\boldsymbol{\eta}_1)\mathbf{B}^{(\mathrm{cur})},\\
&\boldsymbol{\tau}^{(\mathrm{new})} \leftarrow (\mathbf{B}^{(\mathrm{new})})^T\mathbf{B}^{(\mathrm{new})},
\end{aligned}\right\} \equiv \boldsymbol{\tau}^{(\mathrm{new})} \leftarrow \boldsymbol{\phi}_{\boldsymbol{\tau}^{(\mathrm{cur})}}(\boldsymbol{\eta}_1),\\
&\mathbf{w} \leftarrow \mathbf{m}.
\end{aligned}$$

In a local coordinate, our update is exactly the Euclidean GD with momentum.

The intermediate step in $\mathbf{B}$ allows us to include structures in $\mathbf{B}$ by considering sparse subgroup structures in $\mathbf{B}$ as suggested by Lin et al. (2021a). This gives rise to structured Gaussian manifolds as we will discuss in B.1. In the structured cases, our update with momentum also enjoys a Lie subgroup invariance (Lin et al., 2021b).

References

Kwangjun Ahn and Suvrit Sra. From Nesterov's estimate sequence to Riemannian acceleration. In *Conference on Learning Theory*, pages 84–118. PMLR, 2020.

Foivos Alimisis, Antonio Orvieto, Gary Bécigneul, and Aurelien Lucchi. A continuous-time perspective for modeling acceleration in Riemannian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1297–1307. PMLR, 2020.

Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Wu Lin, Frank Nielsen, Khan Mohammad Emtiyaz, and Mark Schmidt. Tractable structured natural-gradient descent using local parameterizations. In *International Conference on Machine Learning*, pages 6680–6691. PMLR, 2021a.

Wu Lin, Frank Nielsen, Mohammad Emtiyaz Khan, and Mark Schmidt. Structured second-order methods via natural gradient descent. *arXiv preprint arXiv:2107.10884*, 2021b.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417, 2015.

Extended Abstract Track

Appendix A. Backgound

A.1. Fisher Information Matrix

Under parametrization $\boldsymbol{\tau}$, the Fisher information matrix is defined as

$$\mathbf{F}_\tau(\boldsymbol{\tau}^{(\text{cur})}) := -E_{q(z|\tau)}[\nabla_\tau^2 \log q(\mathbf{z}|\boldsymbol{\tau})], \tag{8}$$

where $q(\mathbf{z}|\boldsymbol{\tau})$ is a probabilistic distribution parameterized by $\boldsymbol{\tau}$ such as the Gaussian distribution with $\boldsymbol{\tau} = \{\boldsymbol{\mu}, \mathbf{S}\}$, where $\boldsymbol{\mu}$ is the mean and $\mathbf{S}$ is the inverse covariance matrix.

A.2. Christoffel Symbols

The Christoffel symbols of the first kind of the Fisher metric connection are defined as

$$\Gamma_{d,ab}(\boldsymbol{\tau}) := \tfrac{1}{2}[\partial_a F_{bd}(\boldsymbol{\tau}) + \partial_b F_{ad}(\boldsymbol{\tau}) - \partial_d F_{ab}(\boldsymbol{\tau})]\Big|_{\tau=r}, \tag{9}$$

where $F_{bd}(\boldsymbol{\tau})$ denotes the (b, d) entry of metric $\mathbf{F}_\tau$ and ∂_b denotes the partial derivative w.r.t. the b-th entry of $\boldsymbol{\tau}$.

The Christoffel symbols of the second kind are defined as $\Gamma^c_{ab}(\boldsymbol{\tau}) := \sum_d F^{cd}(\boldsymbol{\tau})\Gamma_{d,ab}(\boldsymbol{\tau})$ where $F^{cd}(\boldsymbol{\tau})$ denotes the (c,d) entry of $\mathbf{F}_\tau^{-1}$. Observe that the Christoffel symbols involves computing all partial derivatives of the Fisher matrix $\mathbf{F}_\tau$.

A.3. Riemannian Exponential Map

The exponential map is defined via a geodesic. The geodesic generalizes the notion of a straight line to a manifold. The geodesic $\mathbf{r}(t)$ satisifies the following second-order nonlinear system of ODEs with initial values $\mathbf{r}(0) = \mathbf{x}$ and $\dot{\mathbf{r}}(0) = \boldsymbol{\nu}$, where $\mathbf{x}$ denotes a point on the manifold and $\boldsymbol{\nu}$ is a Riemannian gradient,

$$\ddot{r}^c(t) + \sum_{a,b} \Gamma^c_{ab}(\mathbf{r}(t))\dot{r}^a(t)\dot{r}^b(t) = 0, \tag{10}$$

where $r^c(t)$ denotes the c-th entry and Γ^c_{ab} is the Christoffel symbol of the second kind.

The Riemannian exponential map is defined as

$$\mathrm{Exp}(\mathbf{x}, \boldsymbol{\nu}) := \mathbf{r}(1), \tag{11}$$

where $\mathbf{x}$ denotes an initial point and $\boldsymbol{\nu}$ is an initial Riemannian gradient so that $\mathbf{r}(0) = \mathbf{x}$ and $\dot{\mathbf{r}}(0) = \boldsymbol{\nu}$.

A.4. Parallel Transport Map for Riemannian Gradients

Given a geodesic curve $\mathbf{r}(t)$, denote by $\mathbf{v}(t)$ a smooth Riemannian gradient field that satisfies the following first-order linear system of ODEs with initial value $\mathbf{v}(0) = \boldsymbol{\nu}$,

$$\dot{v}^c(t) + \sum_{a,b} \Gamma^c_{ab}(\mathbf{r}(t))v^a(t)\dot{r}^b(t) = 0. \tag{12}$$

Extended Abstract Track

The transport map $\hat{\Gamma}_{\boldsymbol{\tau}^{(\mathrm{old})} \to \boldsymbol{\tau}^{(\mathrm{old})}}(\boldsymbol{\nu})$ transports the Riemannian gradient $\boldsymbol{\nu}$ at $\boldsymbol{\tau}^{(\mathrm{old})}$ to $\boldsymbol{\tau}^{(\mathrm{new})}$ as follows,

$$\hat{\Gamma}_{\boldsymbol{\tau}^{(\mathrm{cur})} \to \boldsymbol{\tau}^{(\mathrm{new})}}(\boldsymbol{\nu}) := \mathbf{v}(1), \tag{13}$$

where $\mathbf{r}(0) = \boldsymbol{\tau}^{(\mathrm{cur})}$, $\mathbf{r}(1) = \boldsymbol{\tau}^{(\mathrm{new})}$, and $\mathbf{v}(0) = \boldsymbol{\nu}$. It can be computationally challenging to solve this linear ODE due to the Christoffel symbol.

A.5. Parallel Transport Map for Euclidean Gradients

Given a geodesic curve $\mathbf{r}(t)$, denote by $\boldsymbol{\omega}(t)$ a smooth Euclidean gradient field on manifold $\mathcal{M}$ that satisfies the following first-order linear system of ODEs with initial value $\boldsymbol{\omega}(0) = \mathbf{m}$,

$$\dot{\omega}_c(t) - \sum_{a,b} \Gamma^a_{cb}(\mathbf{r}(t))\omega_a(t)\dot{r}^b(t) = 0. \tag{14}$$

The transport map transports the Euclidean gradient $\mathbf{g}$ at $\boldsymbol{\tau}^{(\mathrm{cur})}$ to $\boldsymbol{\tau}^{(\mathrm{new})}$ as shown below,

$$\Gamma_{\boldsymbol{\tau}^{(\mathrm{cur})} \to \boldsymbol{\tau}^{(\mathrm{new})}}(\mathbf{g}) := \boldsymbol{\omega}(1), \tag{15}$$

where $\mathbf{r}(0) = \boldsymbol{\tau}^{(\mathrm{cur})}$, $\mathbf{r}(1) = \boldsymbol{\tau}^{(\mathrm{new})}$, and $\boldsymbol{\omega}(0) = \mathbf{g}$.

These two transport maps are related as follows,

$$\hat{\Gamma}_{\boldsymbol{\tau}^{(\mathrm{cur})} \to \boldsymbol{\tau}^{(\mathrm{new})}}(\mathbf{F}_{\boldsymbol{\tau}}^{-1}(\boldsymbol{\tau}^{(\mathrm{cur})})\mathbf{g}) = \mathbf{F}_{\boldsymbol{\tau}}^{-1}(\boldsymbol{\tau}^{(\mathrm{new})})\Gamma_{\boldsymbol{\tau}^{(\mathrm{cur})} \to \boldsymbol{\tau}^{(\mathrm{new})}}(\mathbf{g}), \tag{16}$$

where $\mathbf{g}$ is a Euclidean gradient evaluated at $\boldsymbol{\tau}^{(\mathrm{cur})}$.

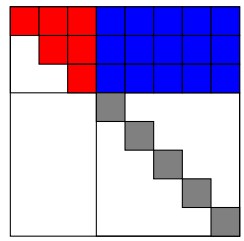


Figure 2: Visualization of the subgroup

A.6. A First-order Approximation of the Transport Map

We can also consider a first-order approximation of the transport map for Euclidean gradients as follows,

$$\Gamma_{\eta_0 \to \eta_1}(\mathbf{m}^{(\eta_0)}) = \boldsymbol{\omega}(1) \approx \boldsymbol{\omega}(0) + \dot{\boldsymbol{\omega}}(0), \tag{17}$$

where we have to evaluate the Christoffel symbols as discussed below.

By the transport ODE in Eq. 14, we can compute $\dot{\boldsymbol{\omega}}(0)$ as follows,

$$\dot{\omega}_c(0) - \sum_{a,b} \Gamma^a_{cb}(\mathbf{r}(0))\omega_a(0)\dot{r}^b(0) = 0,$$

where $\mathbf{r}(0) = \boldsymbol{\eta}_0$ is the current point and $\dot{\mathbf{r}}(0)$ is the Riemannian gradient so that $\boldsymbol{\eta}_1 = \mathrm{Exp}(\boldsymbol{\eta}_0, \dot{\mathbf{r}}(0))$. In our case, $\dot{\mathbf{r}}(0) = -\sqrt{\alpha}\mathbf{F}_\eta^{-1}(\eta_0)\mathbf{m}^{(\eta_0)}$ and $\boldsymbol{\omega}(0) = \mathbf{m}^{(\eta_0)}$.

Extended Abstract Track

Note that the Fisher matrix and Christoffel symbols are evaluated at $\boldsymbol{\eta}_0 = \mathbf{0}$. Thus, the computation can be simplified since $\boldsymbol{\eta}$ is a generalized Riemannian normal coordinate.

A.7. Update 5 is Equivalent to Alimisis et al. (2020)

The update of Alimisis et al. (2020) is defined as

$$\begin{aligned}
\boldsymbol{\nu}^{(\mathrm{cur})} &\leftarrow \beta \mathbf{z}^{(\mathrm{cur})} + \sqrt{\alpha}\mathbf{F}_\tau^{-1}(\boldsymbol{\tau}^{(\mathrm{cur})})\mathbf{g}(\boldsymbol{\tau}^{(\mathrm{cur})}), \\
\boldsymbol{\tau}^{(\mathrm{new})} &\leftarrow \mathrm{Exp}(\boldsymbol{\tau}^{(\mathrm{cur})}, -\sqrt{\alpha}\boldsymbol{\nu}^{(\mathrm{cur})}), \\
\mathbf{z}^{(\mathrm{new})} &\leftarrow \hat{\Gamma}_{\tau^{(\mathrm{cur})} \to \tau^{(\mathrm{new})}}(\boldsymbol{\nu}^{(\mathrm{cur})}),
\end{aligned} \tag{18}$$

where $\boldsymbol{\nu}$ is a Riemannian gradient, and $\hat{\Gamma}_{\tau^{(\mathrm{cur})} \to \tau^{(\mathrm{new})}}$ is the parallel transport map for Riemannian gradients.

Due to Eq. 16, update 18 and update 5 are equivalent by the observation that $\mathbf{m}^{(\mathrm{cur})} = \mathbf{F}(\boldsymbol{\tau}^{(\mathrm{cur})})\boldsymbol{\nu}^{(\mathrm{cur})}$ and $\mathbf{w}^{(\mathrm{new})} = \mathbf{F}(\boldsymbol{\tau}^{(\mathrm{new})})\mathbf{z}^{(\mathrm{new})}$.

Appendix B. More Examples

B.1. Structured Gaussian with Constant Mean

Consider a d-dimensional Gaussian family with zero mean $q(\mathbf{z}|\boldsymbol{\tau})$, where the global parametrization $\boldsymbol{\tau}$ is the inverse covariance, which is symmetric positive-definite.

We introduce an auxiliary parameterization $\mathbf{B}$ such that $\boldsymbol{\tau} = \mathbf{B}^T\mathbf{B}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{B}_A & \mathbf{B}_C \\ \mathbf{0} & \mathbf{B}_D \end{bmatrix}$, where $\mathbf{B}_A$ is an invertible upper-triangular matrix and $\mathbf{B}_D$ is an invertible diagonal matrix. The auxiliary parameter $\mathbf{B}$ lives in a subgroup of the upper-triangular group illustrated in Figure 2.

We first define a constant matrix $\mathbf{D} = \begin{bmatrix} \frac{\mathrm{Tril}(\mathbf{1}_A)}{\sqrt{2}} & \mathbf{1}_B \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I}_D \end{bmatrix}$, where $\mathbf{1}$ is the matrix of all ones, $\mathbf{I}_D$ is the identity matrix, Tril() extracts the upper-triangular half. We define a local parameterization $\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_A & \boldsymbol{\eta}_C \\ \mathbf{0} & \boldsymbol{\eta}_D \end{bmatrix}$ associated to $\mathbf{B}^{(\mathrm{cur})}$ as $\mathbf{B} = \mathbf{h}(\mathbf{D} \odot \boldsymbol{\eta})\mathbf{B}^{(\mathrm{cur})}$, where $\odot$ is the element-wise product, $\mathbf{h}(\mathbf{N})$ is defined in the main text, $\boldsymbol{\eta}_A$ is an upper-triangular matrix, $\boldsymbol{\eta}_D$ is a diagonal matrix. In this case, the local parameter $\boldsymbol{\eta}$ lives in the Lie algebra of $\mathbf{B}$.

We can show that $\boldsymbol{\eta}$ is a generalized Riemannian normal coordinate associated to $\boldsymbol{\tau}^{(\mathrm{cur})}$ as discussed in the main text. Moreover, $\mathbf{F}_\eta(\boldsymbol{\eta}_0) = \mathbf{I}$ where $\boldsymbol{\eta}_0 = \mathbf{0}$.

The Christoffel symbols at $\boldsymbol{\eta}_0$ do not vanish. Since they only are required if we use the first-order approximation of the transport map, we will not include the derivation in the appendices. However, they can be efficiently computed.

Notice that $\boldsymbol{\tau} = (\mathbf{B}^{(\mathrm{cur})})^T\mathbf{h}^T(\boldsymbol{\eta})\mathbf{h}(\boldsymbol{\eta})\mathbf{B}^{(\mathrm{cur})} = (\mathbf{B}^{(\mathrm{new})})^T\mathbf{h}^T(\boldsymbol{\xi})\mathbf{h}(\boldsymbol{\xi})\mathbf{B}^{(\mathrm{new})}$ and $\mathbf{B}^{(\mathrm{new})} = \mathbf{h}(\boldsymbol{\eta}_1)\mathbf{B}^{(\mathrm{cur})}$. We have $\mathbf{h}^T(\boldsymbol{\eta})\mathbf{h}(\boldsymbol{\eta}) = \mathbf{h}^T(\boldsymbol{\eta}_1)\mathbf{h}^T(\boldsymbol{\xi})\mathbf{h}(\boldsymbol{\xi})\mathbf{h}(\boldsymbol{\eta}_1)$. Due to the structure restriction on $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$, $\mathbf{h}(\boldsymbol{\eta})$ and $\mathbf{h}(\boldsymbol{\xi})$ cannot be an orthogonal matrix. Thus, $\mathbf{h}(\boldsymbol{\eta}) = \mathbf{h}(\boldsymbol{\xi})\mathbf{h}(\boldsymbol{\eta}_1)$. If $\mathbf{h}$ is the matrix exponential denoted by Expm(), we have $\boldsymbol{\eta} = \mathrm{Expm}^{-1}\big(\mathrm{Expm}(\boldsymbol{\xi})\mathrm{Expm}(\boldsymbol{\eta}_1)\big)$. By the Baker–Campbell–Hausdorff formula, we have $\boldsymbol{\eta} = \boldsymbol{\xi} + \boldsymbol{\eta}_1 + \frac{1}{2}[\boldsymbol{\xi}, \boldsymbol{\eta}_1] + O(\boldsymbol{\xi}^2)$, where $[\boldsymbol{\xi}, \boldsymbol{\eta}_1] := \boldsymbol{\xi}\boldsymbol{\eta}_1 - \boldsymbol{\eta}_1\boldsymbol{\xi}$ is the Lie bracket (A.K.A., the matrix commutator).

Extended Abstract Track

The Jacobian-vector product at $\boldsymbol{\xi}_0$ can be simplified as

$$\mathbf{J}(\boldsymbol{\xi}_0)\mathbf{w} = \frac{\partial \mathrm{Tr}(\mathbf{w}^T\boldsymbol{\eta})}{\partial \boldsymbol{\xi}}\Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0} = \mathbf{w} + \frac{1}{2}(\mathbf{w}\boldsymbol{\eta}_1^T - \boldsymbol{\eta}_1^T\mathbf{w}).$$

We can similarly obtain the proposed update.

Appendix C. Numerical Results

C.1. Optimization on Symmetric Positive-definite Matrices

We consider manifold optimization on positive-definite matrices $\mathcal{S}_{++}^{d\times d}$, where the Riemannian exponential and transport maps admit a closed-form expression. However, our approach can be extended to structured cases, where these maps do not have a closed-form expression.

We evaluate the proposed method in an example of the metric nearness problem considered in Lin et al. (2021a). The objective function is $\ell(\mathbf{Z}) = \frac{1}{2N}\sum_{i=1}^N \|\mathbf{Z}\mathbf{Q}\mathbf{x}_i - \mathbf{x}_i\|_2^2$, where $\mathbf{x}_i \in \mathcal{R}^d$, $\mathbf{Q} \in \mathcal{S}_{++}^{d\times d}$ and $\mathbf{Z} \in \mathcal{S}_{++}^{d\times d}$. The optimal is $\mathbf{Q}^{-1}$. We randomly generate $\mathbf{x}_i$ and $\mathbf{Q}$ with $d = 100$, $N_{\mathrm{train}} = 5 \times 10^5$ for training and $N_{\mathrm{test}} = 10^5$ for testing. All methods are trained using mini-batches, where the size of mini-batch is 100. We also consider existing Riemannian momentum methods such as Ahn and Sra (2020) and Alimisis et al. (2020) as baselines. From Figure 1(a), we can see that the proposed method using a local coordinate performs as good as existing Riemannian momentum methods using a global coordinate with the exact exponential and transport maps.

C.2. Unconstrained Optimization

We consider the following valley-shaped function suggested by in Lin et al. (2021a). Rosenbrock : $\ell(\mathbf{z}) = \frac{1}{d}\sum_{i=1}^{d-1}\left[100(z_{i+1} - z_i)^2 + (z_i - 1)^2\right]$, where $d = 10^3$. Our baselines are the BFGS provided by SciPy and the method of Lin et al. (2021a). Our method and the method of Lin et al. (2021a) use a structured Hessian information, which can be efficiently computed via a Hessian-vector product. We consider a block Heisenberg structure as suggested by Lin et al. (2021a), where our update introduce momentum into the method of Lin et al. (2021a). From Figure 1(b), we can see that the momentum can improve the performance.

C.3. Deep Learning

We consider a CNN model with the VGG-16 neural network architecture. We train the model with matrix Gaussian (Lin et al., 2021a) for each layer-wise matrix weights on dataset"CIFAR-10". For each Gaussian, we use a Kronecker product of two dense groups. The iteration cost and the structure is similar to the KFAC approximation (Martens and Grosse, 2015). We also use a similar approximation as KFAC to compute natural gradients. We include the KFAC method as our baseline method. We train the model with mini-batch size 256. From Figure 1(c), we can see that the momentum can improve the performance and our method performs similarly to KFAC.

However, our approach enables us to introduce more sparse structures along with the Kronecker structure such as a Kronecker product of two sparse groups, which can further reduce the iteration cost of our method. We are currently investigating sparse updates that can further reduce the iteration cost while maintaining the performance.

Extended Abstract Track