# RIEMANNIAN DIFFUSION ADAPTATION OVER GRAPHS WITH APPLICATION TO ONLINE DISTRIBUTED PCA

*Xiuheng Wang* [*], *Ricardo Augusto Borsoi* [†], *Cédric Richard* [*]

[*] Université Côte d'Azur, CNRS, OCA, France
[†] Université de Lorraine, CNRS, CRAN, Vandoeuvre-lès-Nancy, France
xiuheng.wang@oca.eu, ricardo.borsoi@univ-lorraine.fr, cedric.richard@unice.fr

## ABSTRACT

Distributed adaptation and learning recently gained considerable attention in solving optimization problems with streaming data collected by multiple agents over a graph. This work focuses on such problems where the solutions lie on a Riemannian manifold. This research topic is of particular interest for many applications, e.g., principal component analysis (PCA). Although several incremental and consensus algorithms have been proposed, there is a lack of methods designed for general Riemannian manifolds with efficient diffusion strategies. In this paper, we devise two Riemannian diffusion adaptation strategies, namely, adaptation-then-combination (ATC) and combination-then-adaptation (CTA), for decentralized Riemannian optimization over graphs. In the adaptation step, a Riemannian stochastic gradient descent method (SGD) is used to estimate the local solution at each node. In the combination step, the local estimates at the different nodes are combined by computing the weighted Fréchet mean over the neighborhood of each node. We apply our algorithms to online distributed PCA and compare them to both non-cooperative and centralized solutions.

***Index Terms***— Diffusion strategy, Riemannian manifolds, distributed optimization, multi-agent system, online.

## 1. INTRODUCTION

Distributed adaptation and learning aims to solve global, stochastic optimization problems by networked agents through local interactions and in the absence of prior knowledge on the probability distributions of measured data [1]. In this work, we consider a collection of $K$ agents over a graph $\mathcal{G}$. At each time instant $t$, each agent $k$ observes one independent realization $\boldsymbol{x}_{k,t}$ of a random streaming data $\boldsymbol{x}_k \in \mathbb{R}^n$. In the decentralized setting, this work deals with the multi-agent optimization problem over a Riemannian manifold $\mathcal{M}$:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathcal{M}} F(\boldsymbol{w}), \tag{1}$$

The work of Cédric Richard was supported in part by the French Government through the 3IA Côte d'Azur Investments in the Future Project under grant ANR-19-P3IA-0002, and in part by grant ANR-19-CE48-0002.

where $F(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} f_k(\boldsymbol{w})$ is a global cost function for the network with $f_k : \mathcal{M} \to \mathbb{R}$ a local risk function defined for each agent by:

$$f_k(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}_k}\left\{q(\boldsymbol{w};\boldsymbol{x}_k)\right\}, \tag{2}$$

in terms of some loss function $q$. The expectation is computed over the unknown distribution of the data $\boldsymbol{x}_k$, which makes it necessary to use a stochastic approximation based on the set of independent realizations $\boldsymbol{x}_{k,t}$, observed sequentially over time. A wide range of applications in machine learning and signal processing can be written in the form of (1), including dictionary learning, PCA, low-rank matrix completion [2].

Distributed adaptation and learning in Euclidean spaces has been extensively studied, including incremental [3], consensus [4] and diffusion [5, 6] strategies. In particular, diffusion strategies have been demonstrated in [7, 8] to offer improved performance and stability guarantees under constant step-size learning and adaptive scenarios, with extensions to constrained [9], multi-task [10, 11], and non-convex [12] environments. Nevertheless, all these decentralized algorithms operate in Euclidean spaces and may fail when dealing with problem (1) defined on $\mathcal{M}$. For a special case of $\mathcal{M}$, one may convert the constraint of $\mathcal{M}$ into a cost function in the Euclidean space and solve (1) in a constrained setting [9]. However, a more general and natural way is to address problem (1) by developing unconstrained optimization on $\mathcal{M}$ [2].

To solve (1), the incremental and consensus strategies have been extended to distributed optimization on specific manifolds, including the unit sphere [13] and Stiefel manifolds [14, 15]. An augmented Lagrangian method [16] was also designed for the Stiefel manifold. However, the diffusion strategy has not been investigated on manifolds though it has been proved with superior properties in Euclidean space [7, 8]. Recently, another distributed method [17] considered a graph filter to process statistics in [18] for streaming data on general manifolds, but it was only designed for change point detection. Decentralized optimization on manifolds was also studied in [19] for the natural gradient descent.

In this work, we introduce two general decentralized Riemannian adaptation and learning methods: the Riemannian ATC and CTA diffusion strategies. Both consist of a generalization of Euclidean ATC and CTA diffusion adaptation

strategies to Riemannian manifolds. Specifically, we employ Riemannian SGD to infer a local solution at each individual node in the adaptation step. In the combination step, we merge the local estimates from various nodes by calculating the weighted Fréchet mean within the neighborhood of each node. Finally, we consider an application for online distributed PCA to demonstrate the effectiveness of our strategies on both synthetic and real data.

## 2. BACKGROUND

In this section, we introduce some basic concepts of Riemannian geometry [2], taking the Grassmann manifold as an example. This example is motivated by the experiments related to online distributed PCA proposed in Section 5.

A *Riemannian manifold* $(\mathcal{M}, g)$ is defined by a constrained set $\mathcal{M}$ equipped with a *Riemannian metric* $g_x(\cdot\,,\cdot) : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$, defined for each $x \in \mathcal{M}$, where $T_x\mathcal{M}$ is called the *tangent space* of $\mathcal{M}$ at $x$. The *geodesic distance* is defined as $d_{\mathcal{M}}(\cdot\,,\cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ and satisfies all conditions to be a metric. The *exponential map* $w = \exp_x(v)$ defines the point $w$ of $\mathcal{M}$ located on the unique geodesic $\gamma_v(t)$ such that $\gamma_v(0) = x$, $\gamma_v'(0) = v$ and $\gamma_v(1) = w$. A *retraction* $R_x : T_x\mathcal{M} \to \mathcal{M}$ is a second-order approximation to the exponential map, satisfying $d_{\mathcal{M}}(R_x(tv), \exp_x(tv)) = O(t^3)$. Let $f : \mathcal{M} \to \mathbb{R}$ be a smooth function. The *Riemannian gradient* of $f$ at $x \in \mathcal{M}$ is defined as the unique tangent vector $\nabla f(x) \in T_x\mathcal{M}$ satisfying $\frac{d}{dt}\big|_{t=0} f(\exp_x(tv)) = \langle \nabla f(x), v\rangle_x$ for all $v \in T_x\mathcal{M}$.

The Grassmann manifold $\mathcal{G}_n^p$, a set of $p$-dimensional linear subspaces of $\mathbb{R}^n$, can be regarded as a smooth quotient manifold of the Stiefel manifold $\mathcal{S}_n^p = \{\boldsymbol{U} \in \mathbb{R}^{n\times p} : \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_p\}$, i.e., $\mathcal{G}_n^p = \mathcal{S}_n^p/\mathcal{O}_p = \{\pi(\boldsymbol{U}) : \boldsymbol{U} \in \mathcal{S}_n^p\}$ where $\mathcal{O}_p = \{\boldsymbol{U} \in \mathbb{R}^{p\times p} : \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_p\}$ is the orthogonal group and $\pi : \mathcal{S}_n^p \to \mathcal{G}_n^p$ is the map $\pi(\boldsymbol{U}) = \{\boldsymbol{U}\boldsymbol{O} : \boldsymbol{O} \in \mathcal{O}_p\}$. The geodesic distance between two subspaces $\pi(\boldsymbol{U}_1)$ and $\pi(\boldsymbol{U}_2)$ of $\mathcal{G}_n^p$, spanned by orthonormal matrices $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, is defined as follows [20]:

$$d_{\mathcal{G}_n^p}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \|\cos^{-1}(\boldsymbol{\theta})\|_2\,, \tag{3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^p$ contains the singular values of $\boldsymbol{U}_1^T\boldsymbol{U}_2$, namely, it is related to its singular value decomposition (SVD) as $\boldsymbol{U}_1^T\boldsymbol{U}_2 = \boldsymbol{V}_1^T\mathrm{diag}(\boldsymbol{\theta})\boldsymbol{V}_2$. Define $\bar{f} : \mathcal{S}_n^p \to \mathbb{R}$, we have $f(\pi(\boldsymbol{U})) = \bar{f}(\boldsymbol{U})$ for all $\pi(\boldsymbol{U}) \in \mathcal{G}_n^p$. The Riemannian gradient $\nabla f$ at $\pi(\boldsymbol{U}) \in \mathcal{G}_n^p$ is given by:

$$\nabla f(\pi(\boldsymbol{U})) = \nabla \bar{f}(\boldsymbol{U}) = \boldsymbol{P}_{\boldsymbol{U}}^{\mathcal{G}_n^p}(\boldsymbol{G})\,, \tag{4}$$

with $\boldsymbol{P}_{\boldsymbol{U}}^{\mathcal{G}_n^p}(\boldsymbol{G}) = (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{G}$, where $\boldsymbol{G} \in \mathbb{R}^{n\times p}$ is the Euclidean gradient of $\bar{f}$ at $\boldsymbol{U}$. Let $\boldsymbol{\xi} \in T_{\pi(\boldsymbol{U})}\mathcal{G}_n^p$, and let $\boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y} = \boldsymbol{U} + \boldsymbol{\xi}$ be the thin SVD of $\boldsymbol{U} + \boldsymbol{\xi} \in \mathbb{R}^{n\times p}$. A numerically stable second-order retraction $R_{\pi(\boldsymbol{U})} : T_{\pi(\boldsymbol{U})}\mathcal{G}_n^p \to \mathcal{G}_n^p$ on $\mathcal{G}_n^p$ is given by [2]

$$R_{\pi(\boldsymbol{U})}(\boldsymbol{\xi}) = \pi\big(\boldsymbol{X}\boldsymbol{Y}^T\big)\,. \tag{5}$$

## 3. RIEMANNIAN CENTRALIZED SOLUTION

Unlike the Euclidean distributed setting, problem (1) is defined on $\mathcal{M}$. Thus, we consider directly using Riemannian optimization tools. To begin with, let us consider the centralized setting where, given a collection of data $\boldsymbol{X}_t = \{\boldsymbol{x}_{k,t}\}_{k=1}^K$ that are realizations of some random variable $\boldsymbol{X} = \{\boldsymbol{x}_k\}_{k=1}^K$, one aims to minimize the global cost function in (1) on $\mathcal{M}$ with a Riemannian SGD algorithm [21] and a step size $\Lambda > 0$, say,

$$\boldsymbol{w}_t = R_{\boldsymbol{w}_{t-1}}\big(-\Lambda H(\boldsymbol{w}_{t-1}, \boldsymbol{X}_{t-1})\big)\,, \tag{6}$$

where $\boldsymbol{w}_t \in \mathcal{M}$ is the solution at time instant $t$, and $H(\boldsymbol{w}, \boldsymbol{X})$ denotes a stochastic approximation of the Riemannian gradient of the loss function satisfying:

$$\mathbb{E}_{\boldsymbol{X}}\big\{H(\boldsymbol{w}, \boldsymbol{X})\big\} = \int H(\boldsymbol{m}, \boldsymbol{X})dP(\boldsymbol{X}) = \nabla F(\boldsymbol{w})\,.$$

Note that algorithm (6) is not distributed and requires access to the data over the entire graph. The system must comprise a fusion center, which initiates the update process described in (6) only after receiving data from all $K$ agents.

## 4. RIEMANNIAN DIFFUSION ADAPTATION

As discussed above, to execute the centralized stochastic solution (6), one would need simultaneous access to information from all nodes of the graph to update a new estimate $\boldsymbol{w}_t$. In the case of Euclidean spaces, this challenge has been tackled by the well-known diffusion adaptation strategies [5, 6], which only require communication across local neighborhoods of the graph. This paper aims to generalize such strategies on Riemannian manifolds to solve (1) using only local interaction and information exchange among network agents.

In the decentralized setting, after initializing the solution $\boldsymbol{w}_{k,0} \in \mathcal{M}$ for all $k$, agent $k$ is particularly interested in solving (1) with the following Riemannian SGD algorithm:

$$\boldsymbol{w}_{k,t} = R_{\boldsymbol{w}_{k,t-1}}\big(-\lambda h(\boldsymbol{w}_{k,t-1}, \boldsymbol{x}_{k,t-1})\big)\,, \tag{7}$$

where $\lambda > 0$ is a step-size parameter and $h(\boldsymbol{w}, \boldsymbol{x}_k)$ denotes a stochastic approximation of the Riemannian gradient of the loss $f_k(\boldsymbol{w})$, satisfying:

$$\mathbb{E}_{\boldsymbol{x}_k}\big\{h(\boldsymbol{w}, \boldsymbol{x}_k)\big\} = \int h(\boldsymbol{w}, \boldsymbol{x}_k)dP(\boldsymbol{x}_k) = \nabla f_k(\boldsymbol{w})\,.$$

A common approach to construct the stochastic approximation $h(\boldsymbol{w}_{k,t}, \boldsymbol{x}_{k,t})$ is to compute the Riemannian gradient of the stochastic approximation of the loss function (2), approximating $\boldsymbol{x}_k$ by the instantaneous realization $\boldsymbol{x}_{k,t}$ at time $t$. Note, however, that the iterative equation (7) does not benefit from exchanging information among neighboring agents.

Several distributed strategies have been proposed to minimize a global cost function defined in Euclidean space in a

fully decentralized manner [3, 4, 5, 6]. The appeal of diffusion strategies [5, 6] arises from their inherent attributes of scalability, robustness, and the facilitation of continuous learning and adaptation in response to drifts in the location of the minimizer. There exist mainly two variations of the diffusion adaptation strategies, namely, the adapt-then-combine (ATC) and the combine-then-adapt (CTA).

In the following, we propose two diffusion-adaptation-based strategies to estimate the solution $\boldsymbol{w}$ in a distributed manner through local adaptation and information exchange on a Riemannian manifold $\mathcal{M}$. The proposed method is based on two steps: an *adaptation* step, where a node-wise Riemannian SGD updates the estimated solution at each node, and a *combination* step, in which the solutions at different nodes are exchanged and combined. The combination step is performed by computing the weighted Fréchet means of the solutions at the neighborhood of each graph node.

More specifically, the Riemannian ATC diffusion strategy for solving (1) takes the following form at each agent $k$:

$$\begin{aligned} \boldsymbol{\psi}_{k,t} &= R_{\boldsymbol{w}_{k,t-1}}\big(-\lambda h(\boldsymbol{w}_{k,t-1}, \boldsymbol{x}_{k,t-1})\big)\,, \\ \boldsymbol{w}_{k,t} &= \arg\min_{\boldsymbol{w}\in\mathcal{M}} \sum_{l\in\mathcal{N}_k} a_{lk}\big\{d^2_{\mathcal{M}}(\boldsymbol{w}, \boldsymbol{\psi}_{l,t})\big\}\,, \end{aligned} \tag{8}$$

where $\mathcal{N}_k$ denotes the set of nodes in the neighborhood of node $k$ (including $k$ itself), and the weighting coefficients $a_{lk}$ are non-negative and add to one over $l \in \mathcal{N}_k$:

$$a_{lk} \geq 0, \quad \sum_{l=1}^{K} a_{lk} = 1, \quad \text{and} \quad a_{lk} = 0 \text{ if } l \notin \mathcal{N}_k\,. \tag{9}$$

The condition (9) means that the matrix $\boldsymbol{A} \triangleq [a_{lk}]$ is left-stochastic. The Riemannian ATC diffusion (8) contains two steps. The first is an adaptation step where agent $k$ uses its own data $\boldsymbol{x}_{k,t-1}$ to update its solution $\boldsymbol{\psi}_{k,t}$. The second step is a combination step where the intermediate estimates $\{\boldsymbol{\psi}_{l,t}\}$ from the neighborhood of agent $k$ are combined according to the weighting coefficients $\{a_{lk}\}$ to obtain the estimate $\boldsymbol{w}_{k,t}$.

A similar implementation can be procured through the reordering of the adaptation and combination steps. In the Riemannian CTA implementation, agent $k$ initially combines the prior estimations of its neighbors to derive the intermediate estimate $\boldsymbol{\psi}_{k,t}$, subsequently applying its data to update this intermediate estimate:

$$\begin{aligned} \boldsymbol{\psi}_{k,t} &= \arg\min_{\boldsymbol{w}\in\mathcal{M}} \sum_{l\in\mathcal{N}_k} a_{lk}\big\{d^2_{\mathcal{M}}(\boldsymbol{w}, \boldsymbol{w}_{l,t-1})\big\}\,, \\ \boldsymbol{w}_{k,t} &= R_{\boldsymbol{\psi}_{k,t}}\big(-\lambda h(\boldsymbol{\psi}_{k,t}, \boldsymbol{x}_{k,t})\big)\,. \end{aligned} \tag{10}$$

The Riemannian ATC and CTA strategies are summarized in Algorithms 1 and 2. In Euclidean spaces, the combination step implemented by diffusion strategies, as described in [5, 6], can be understood as the computation of a weighted mean from the neighborhood of agent $k$. However, when dealing with manifold-valued data, we propose to generalize the solution of the combination step in Riemannian ATC (8) and CTA (10) as a *weighted Fréchet mean* [22], a minimum of the expected variance of the Riemannian distance $d_{\mathcal{M}}$. The weighted Fréchet mean serves as a generalization of the center of mass in Euclidean domains, to a manifold $\mathcal{M}$. To compute this weighted Fréchet mean on $\mathcal{M}$, one can employ various techniques such as Riemannian optimization [2] or recursive estimation [23]. In this work, we considered a Riemannian steepest descent algorithm with $L$ iterations.

**Algorithm 1** Riemannian ATC diffusion

Initialize $\{\boldsymbol{w}_{k,-1}\}$ for all $k$ with a random point on $\mathcal{M}$. Given coefficients $\{a_{lk}\}$ satisfying (9), for each time $t \geq 0$ and for each node $k$, repeat:

$\boldsymbol{\psi}_{k,t} = R_{\boldsymbol{w}_{k,t-1}}\big(-\lambda h(\boldsymbol{w}_{k,t-1}, \boldsymbol{x}_{k,t-1})\big)$,
$\boldsymbol{w}_{k,t} = \arg\min_{\boldsymbol{w}\in\mathcal{M}} \sum_{l\in\mathcal{N}_k} a_{lk}\big\{d^2_{\mathcal{M}}(\boldsymbol{w}, \boldsymbol{\psi}_{l,t})\big\}$.

**Algorithm 2** Riemannian CTA diffusion

Initialize $\{\boldsymbol{w}_{k,-1}\}$ for all $k$ with a random point on $\mathcal{M}$. Given coefficients $\{a_{lk}\}$ satisfying (9), for each time $t \geq 0$ and for each node $k$, repeat:

$\boldsymbol{\psi}_{k,t} = \arg\min_{\boldsymbol{w}\in\mathcal{M}} \sum_{l\in\mathcal{N}_k} a_{lk}\big\{d^2_{\mathcal{M}}(\boldsymbol{w}, \boldsymbol{w}_{l,t-1})\big\}$,
$\boldsymbol{w}_{k,t} = R_{\boldsymbol{\psi}_{k,t}}\big(-\lambda h(\boldsymbol{\psi}_{k,t}, \boldsymbol{x}_{k,t})\big)$.

## 5. NUMERICAL EXPERIMENTS

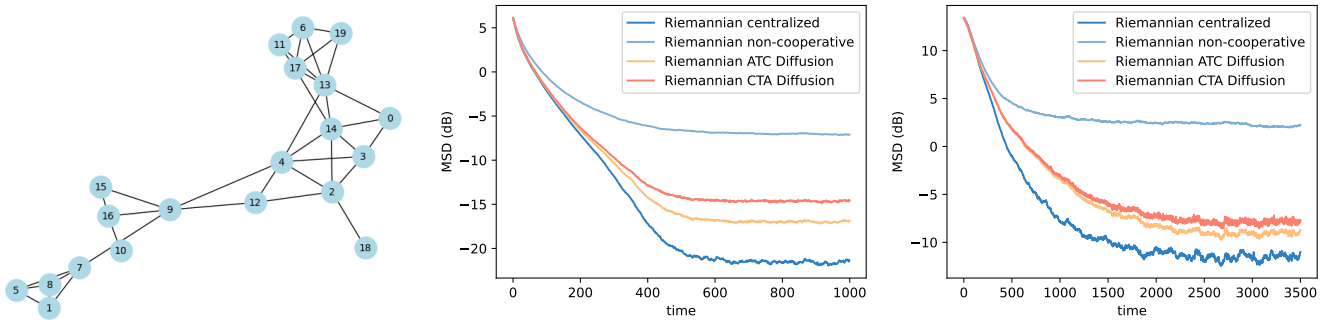
We considered applying our algorithms to the online PCA problem with $\boldsymbol{x}_k \in \mathbb{R}^n$ being data samples observed by each agent $k$. For the global estimation, the online centralized PCA problem can be defined as [24]:

$$\min_{\pi(\boldsymbol{U})\in\mathcal{G}_n^p} -\frac{1}{K}\,\mathbb{E}_{\boldsymbol{X}}\big\{tr(\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U})\big\}\,, \tag{11}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K]$ is the collection of data over all agents and $\pi(\boldsymbol{U})$ represents the global estimate. Note that though various works formulate PCA on the Stiefel manifold [13, 14, 15, 16], the loss function in (11) is invariant to orthonormal transformations. Thus, we formulated the problem on the Grassmannian manifold since it makes the solution unique [24]. In the decentralized setting, we considered the following problem:

$$\min_{\pi(\boldsymbol{U}_k)\in\mathcal{G}_n^p} -\mathbb{E}_{\boldsymbol{x}_k}\big\{tr(\boldsymbol{U}_k^T\boldsymbol{x}_k\boldsymbol{x}_k^T\boldsymbol{U}_k)\big\}\,, \tag{12}$$

where $\pi(\boldsymbol{U}_k)$ represents the local estimate at agent $k$. In the online setting, the expectation in the loss functions of (11) and (12) was approximated by realizations $\boldsymbol{x}_{k,t}$ and $\boldsymbol{X}_t$ at each time instant $t$.




(a) Graph topology. (b) MSD results on synthetic data. (c) MSD results on real data.

**Fig. 1**: Illustration of the graph topology and MSD performance of the algorithms on synthetic and real data.

The Riemannian gradient of the stochastic approximation of the loss function in (12) on the Grassmann manifold was computed using the Euclidean gradient and (4), leading to:

$$h(\boldsymbol{U}_{k,t}, \boldsymbol{x}_{k,t}) = 2(\boldsymbol{I} - \boldsymbol{U}_{k,t}\boldsymbol{U}_{k,t}^T)\boldsymbol{x}_{k,t}\boldsymbol{x}_{k,t}^T\boldsymbol{U}_{k,t}\,. \qquad (13)$$

The Riemannian gradient of the loss function for the centralized problem in (11) was computed similarly. The retraction and the geodesic distance used in the centralized, ATC and CTA strategies in (6), (8) and (10) are defined in (5) and (3), respectively. In order to evaluate the accuracy of the solutions, we considered the Grassmann distance (3) between the estimates at each time instant $\pi(\boldsymbol{U}_{k,t})$ and the optimal solution $\pi(\boldsymbol{U}^*)$, and we defined the mean square deviation (MSD) accordingly as $(1/K)\sum_{k=1}^{K}\mathbb{E}\{d^2_{\mathcal{G}_n^p}(\boldsymbol{U}_{k,t}, \boldsymbol{U}^*)\}$. Similar MSD definitions were used for the centralized and non-cooperative solutions. We considered both synthetic and real data to demonstrate the effectiveness of our strategy. Figure 1a illustrates the graph topology used for simulations, which includes a total of $K = 20$ agents. We selected the weights in matrix $\boldsymbol{A}$ with Metropolis rule [25] and $L = 10$ iterations in the combination step.

### 5.1. Synthetic data

We generated synthetic data as in [14]. First, we set $n = 10$, $p = 5$ and independently sampled $1000K$ data points according to the multivariate Gaussian model to obtain a matrix $\boldsymbol{S} \in \mathbb{R}^{n\times 1000K}$. Let $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be its truncated SVD. We modified the distribution of $\boldsymbol{\Sigma}$ as $\boldsymbol{\Sigma}' = \text{diag}(\sigma^i)$ with $\sigma = 0.8$ and $i = 0, \cdots, n-1$ to reset $\boldsymbol{S}$ as $\boldsymbol{S}' = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{V}^T$. We randomly shuffled and split the columns of $\boldsymbol{S}' \in \mathbb{R}^{n\times 1000K}$ into 1000 subsets to obtain $\boldsymbol{X}_t$ for all time instants $t = 1, \ldots, 1000$. The simulations used fixed step sizes $\Lambda = \lambda = 0.08$, and the MSD results were averaged over 100 times independent Monte Carlo experiments.

Fig. 1b shows the MSD learning curves for Riemannian centralized (6), non-cooperative (7), ATC diffusion (8), and CTA diffusion (10). It can be seen that both Riemannian ATC and CTA diffusion strategies achieved a significant improvement in MSD performance compared to the non-cooperative case, which indicates the benefit of information exchange. The centralized solution achieved the lowest MSD, as it can access information over the whole graph at every iteration. The Riemannian ATC and CTA diffusion algorithms reached intermediate performance. The ATC strategy showed slightly lower MSD, but both approaches yielded similar performance after their combination step, which reduced the estimation variance. This is in agreement with behavior observed in diffusion adaptation algorithms on Euclidean spaces [5, 6].

### 5.2. Real data

We also obtained numerical results on the MNIST dataset [26]. The dataset contains 70000 hand-written images with $n = 784$ pixels. The data matrix was normalized such that the elements are in the range $[0, 1]$ and then centered. We randomly shuffled the images, partitioned them into $K = 20$ subsets, and then ran the algorithms to compute the first $p = 10$ principal components with the fixed step sizes $\Lambda = \lambda = 0.004$. The MSD of the different methods, shown in Fig. 1c, behaves similarly as in the experiment with synthetic data, showing the same relative differences between the performances of the different approaches. However, we note that the MSD values between the two experiments are not directly comparable since the number of components $p$ was different.

## 6. CONCLUSION

In this paper, Riemannian diffusion adaptation over graphs was proposed with application to online distributed PCA. The strategy consisted of two steps, an adaptation step where Riemannian SGD was used to estimate the solution on the manifold at each node, and a combination step where these estimates were combined by computing their weighted Fréchet means over the neighborhood of each graph node. Two approaches were proposed, namely, Riemannian ATC and CTA diffusion. Experimental results on both synthetic and real data demonstrated the efficacy of the proposed strategy. Future work will investigate the performance analysis of the algorithm and further experimental comparisons.

## 7. REFERENCES


[1] A. H. Sayed et al., "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[2] N. Boumal, *An introduction to optimization on smooth manifolds*, Cambridge University Press, 2023.

[3] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, 1997.

[4] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[5] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2009.

[6] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[7] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[8] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[9] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3924–3938, 2014.

[10] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[11] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.

[12] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments—Part I: Agreement at a linear rate," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1242–1256, 2021.

[13] X. Wang, Y. Jiao, H.-T. Wai, and Y. Gu, "Incremental aggregated Riemannian gradient method for distributed PCA," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 7492–7510.

[14] S. Chen, A. Garcia, M. Hong, and S. Shahrampour, "Decentralized Riemannian gradient descent on the Stiefel manifold," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1594–1605.

[15] J. Wang, J. Hu, S. Chen, Z. Deng, and A. M.-C. So, "Decentralized weakly convex optimization over the Stiefel manifold," *arXiv preprint arXiv:2303.17779*, 2023.

[16] L. Wang and X. Liu, "Decentralized optimization over the Stiefel manifold by an approximate augmented Lagrangian function," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3029–3041, 2022.

[17] X. Wang, R. Borsoi, C. Richard, and A. Ferrari, "Distributed change point detection in streaming manifold-valued signals over graphs," in *Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2023.

[18] X. Wang, R. Borsoi, and C. Richard, "Online change point detection on riemannian manifolds with Karcher mean estimates," in *IEEE European Signal Processing Conference (EUSIPCO)*, 2023.

[19] A. Haider, C. Zhang, F. L. Kreyssig, and P. C. Woodland, "A distributed optimisation framework combining natural gradient with Hessian-free for discriminative sequence training," *Neural Networks*, vol. 143, pp. 537–549, 2021.

[20] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[21] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[22] M. Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié," in *Annales de l'institut Henri Poincaré*, 1948, vol. 10, pp. 215–310.

[23] R. Chakraborty and B. C. Vemuri, "Statistics on the Stiefel manifold: Theory and applications," *The Annals of Statistics*, vol. 47, no. 1, pp. 415 – 438, 2019.

[24] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[25] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proceedings of the 5th international conference on Information processing in sensor networks*, 2006, pp. 168–176.

[26] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.